EMBRY-RIDDLE
Aeronautical University
SCHOLARLY COMMONS

Theses - Daytona Beach | Dissertations and Theses

2005

Control System Design and Simulation of Spacecraft Formations

Daniel Dyer
Embry-Riddle Aeronautical University - Daytona Beach

Follow this and additional works at: https://commons.erau.edu/db-theses

 Part of the Space Vehicles Commons

Scholarly Commons Citation

Dyer, Daniel, "Control System Design and Simulation of Spacecraft Formations" (2005). *Theses - Daytona Beach*. 50.
https://commons.erau.edu/db-theses/50

This thesis is brought to you for free and open access by Embry-Riddle Aeronautical University – Daytona Beach at ERAU Scholarly Commons. It has been accepted for inclusion in the Theses - Daytona Beach collection by an authorized administrator of ERAU Scholarly Commons. For more information, please contact commons@erau.edu.

CONTROL SYSTEM DESIGN AND SIMULATION OF SPACECRAFT FORMATIONS

By
Daniel Dyer

A thesis submitted to the Engineering Physics Department
In Partial Fulfillment of the Requirements of
Master of Science in Space Science

Embry-Riddle Aeronautical University
Daytona Beach, FL 32117
2005

UMI Number: EP32048

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.
In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI®

UMI Microform EP32048
Copyright 2011 by ProQuest LLC
All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Copyright by Daniel Dyer 2005
All Rights Reserved

CONTROL SYSTEM DESIGN AND SIMULATION OF

SPACECRAFT FORMATIONS

This thesis was prepared under the direction of the candidate's thesis committee chair, Dr. Mahmut Reyhanoglu, Department of Physical Sciences, and has been approved by the members of his thesis committee. It was submitted to the Department of Physical Sciences and was accepted in partial fulfillment of the requirements for the Degree of

Master of Science in Space Sciences

THESIS COMMITTEE:

M. Reyhanoglu
Dr. Mahmut Reyhanoglu, Chair

Dr. Bereket Berhane, Member

Dr. Tony Hagar, Member

MSSPS Graduate Program Coordinator

Department Chair, Physical Sciences

Associate Chancellor, Daytona Beach Campus

6-27-05
Date

ACKNOWLEDGEMENTS

A special thanks to my parents Bonie and Neil Herberger, my thesis advisor Dr. Mahmut Reyhanoglu, and the Engineering Physics faculty at Embry-Riddle Aeronautical University for their patience and wisdom. Also, to my friends who made the graduate experience even better: Casey, Josh, Justin and Matthew. Finally, in loving memory of my brother Matthew Dyer.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Author:	Daniel Dyer
Title:	Control System Design and Simulation of Spacecraft Formations
Institution:	Embry-Riddle Aeronautical University
Degree:	Master of Space Science
Year:	2005

The objective of this thesis is to analyze an effective control scheme for a formation of spacecraft, implement that control, and provide a 3-D simulation. The thesis first summarizes the progress made in formation flight control schemes and then provides a theoretical framework for the control system design. Using established control design techniques, feedback laws are constructed to control both rotational and translational motion of a group of spacecraft. Computer simulations are carried out using Matlab to generate the ephemeris and attitude data, which are exported to Satellite Tool Kit (STK) to create animations. The process by which Matlab is used to produce the data and how the data are used to produce a 3-D animation in STK is explained. Finally, various formation scenarios are modeled using this same process.

CHAPTER I

1 INTRODUCTION

1.1 Spacecraft Formation Flight

Microsatellites and nanosatellites have become a viable alternative to large expensive spacecraft of the past. However, some projects still require a generally large or complex system that cannot be miniaturized into one small spacecraft. To address this problem, many researchers have proposed that a small "cluster" of spacecraft or "formations" could fulfill a range of necessary tasks similar to the tasks of larger spacecraft. The difference between spacecraft constellations should be mentioned at this point. In a spacecraft constellation no active control is used to keep the group in order, such as GPS, while a formation requires active control to maintain its shape. The benefits of using spacecraft formations are numerous. A few of the many benefits include:

1. Military/scientific applications (e.g. space based interferometers [1], [4], [7]).
2. Potentially lower weight and cost to launch.
3. Faster activation - full formation can be launched at one time.
4. Efficiency in production - common spacecraft could be mass-produced.
5. Increased survivability - loss of single spacecraft does not necessarily ruin group.
6. Easier replacement of individual spacecraft.
7. Flexible system structure – formation can take on multiple shapes.

Essentially, spacecraft formations are becoming an increasingly important approach to many problems due to their low cost and robustness. The popularity and effectiveness of this approach has spawned a need for control schemes to help manage these spacecraft formations.

Many papers have been published concerning control laws for formation flying spacecraft. A popular approach to control law design is based on leader following or nearest neighbor tracking ([9], [18], [19]). Some papers focus on rotational control only ([3], [8], [9]), or on translational control only ([5], [7], [11], [15], [16], [17], [22], [23]), while others combine both rotational and translational control ([2], [10], [12], [14], [21]). A comprehensive survey of spacecraft formation flying guidance can be found in [13].

The control schemes in this thesis will generally be based on methods outlined in [21] and [22]. In [21], a quadratic stability method is employed for translational control and quaternion feedback is proposed for rotational control. In an effort to produce a robust combination of translational and rotational control in this thesis, a modified version of the rotational control approach of [21] is combined with the translational control method described in [22]. The use of quaternions to define the attitude of spacecraft in formation is well described in [6] and [20]. Both rotation and translation of formations are numerically simulated concurrently in order to produce attitude and ephemeris data.

1.2 Contribution of Thesis

This thesis presents an effective control scheme for spacecraft formations and provides a 3-D simulation. In addition, reconfiguration maneuvers are simulated in order to demonstrate the variety of possible formation shapes. The thesis first summarizes the progress made in formation flight control schemes and then provides a framework for the control system. Using established control techniques, methods are constructed to control both rotational and translational motion of a group of spacecraft. A new quaternion

feedback control law is constructed using Lyapunov's second method. Computer simulations are carried out using Matlab to generate the ephemeris and attitude data, and then exported to Satellite Tool Kit (STK). The process by which Matlab is used to produce the data and how the data are used to produce a 3-D animation in STK is explained. Finally, the reconfiguration scenarios are modeled using this same process.

1.3 Organization of Thesis

The organization of the thesis is as follows: Chapter 2 summarizes the basics of Lyapunov's stability theory. In Chapter 3, the translational dynamics of a leader-follower pair are described and a translational control law is derived. Chapter 4 is devoted to rotational dynamics and control. A quaternion feedback law is proposed for eigenaxis rotational maneuvers of a leader-follower pair. In Chapter 5, Matlab simulations of spacecraft formations are presented. STK implementation of the simulation results is explained. Finally, conclusions and future research areas are addressed in Chapter 6 and the enclosed CD contains animations created in STK.

CHAPTER II

2 BACKGROUND ON LYAPUNOV STABILITY THEORY

2.1 Introduction to Lyapunov's Stability Theory

One of Aleksandr Lyapunov's main contributions to control theory involves his method of determining stability of nonlinear systems. Lyapunov's stability criteria and theorems play a role in both the translational and rotational control schemes developed in this thesis. In developing these control schemes, Lyapunov's direct (or second) stability theorem is used to prove that each control law is effective. This chapter briefly describes Lyapunov's stability criteria and summarizes the results on Lyapunov's second stability method. For full details on Lyapunov's stability theory, see [20].

Let $\boldsymbol{x} = (x_1, \cdots, x_n)^T$ denote an n dimensional state vector and consider an autonomous nonlinear dynamical system written in the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) \tag{2.1}$$

where the $\boldsymbol{f}(\boldsymbol{x})$ function is considered to be continuously differentiable. In this thesis an "overdot" represents differentiation with respect to time, i.e. $\dot{\boldsymbol{x}} \triangleq d\boldsymbol{x}/dt$. Let $\boldsymbol{x}_e$ denote an equilibrium state defined as

$$\boldsymbol{f}(\boldsymbol{x}_e) = 0 \tag{2.2}$$

- The equilibrium state $\boldsymbol{x}_e$ is said to be *Lyapunov stable* if for any $\varepsilon > 0$ there exists a real positive number $\delta(\varepsilon, t_0)$ such that

$$\|\boldsymbol{x}(t_0) - \boldsymbol{x}_e\| \le \delta(\varepsilon, t_0) \Rightarrow \|\boldsymbol{x}(t) - \boldsymbol{x}_e\| \le \varepsilon$$

for all $t \geq t_0$ where $\|\mathbf{x}\| \equiv \sqrt{\mathbf{x}^T \mathbf{x}}$.

- The equilibrium state x_e is said to be *locally asymptotically stable* if it is *Lyapunov stable* as explained above and if

$$\|x(t_0) - x_e\| \leq \delta \Rightarrow x(t) \to x_e$$

as $t \to \infty$.

Finally, the equilibrium point x_e is said to be *globally asymptotically stable* if both of the above conditions are met for *any* initial conditions $x(t_0)$. Essentially, if it can be shown that the control laws presented here provide global asymptotic stability, then starting from *any* initial condition the system will reach the desired equilibrium state.

Proving stability of nonlinear systems with the basic stability definitions and without resorting to local approximations can be quite tedious and difficult. Lyapunov's direct method provides a tool to make rigorous, analytical stability claims of nonlinear systems by studying the behavior of a scalar, energy-like Lyapunov function.

Let $E(x)$ be a continuously differentiable function defined on a domain $D \subset \mathbb{R}^n$, which contains the equilibrium state. Then we have the following definitions:

- $E(x)$ is said to be positive definite if $E(x_e) = 0$ and

$$E(x_e) > 0 \quad \forall \quad x \in D - \{x_e\}$$

- $E(\boldsymbol{x})$ is positive semidefinite in the same domain if

$$E(\boldsymbol{x}) \geq 0 \ \forall \ \boldsymbol{x} \in D$$

Negative definite and negative semidefinite are defined as: if $-E$ is positive definite or if $-E$ is positive semidefinite, respectively.

2.2 Lyapunov's Second Stability Theorem

Consider the dynamical system (2.1) and assume that $\boldsymbol{x}_e$ is an isolated equilibrium state. If a positive-definite scalar function $E(\boldsymbol{x})$ exists in a region D around the equilibrium state $\boldsymbol{x}_e$, with continuous first partial derivatives with respect to, where the following conditions are met:

1) $E(\boldsymbol{x}) > 0$ for all $\boldsymbol{x} \neq \boldsymbol{x}_e$ in D, $E(\boldsymbol{x}_e) = 0$.

2) $\dot{E}(\boldsymbol{x}) \leq 0$ for all $\boldsymbol{x} \neq \boldsymbol{x}_e$ in D.

then the equilibrium point is *stable.*

If, in addition to 1 and 2,

3) $\dot{E}(\boldsymbol{x})$ is not identically zero along any solution of (2.1) *other* than $\boldsymbol{x}_e$, then the equilibrium point is *locally asymptotically stable.*

If, in addition to 3,

4) there exists in the entire state space a positive-definite function $E(\boldsymbol{x})$ which is radially unbounded; i.e., $E(\boldsymbol{x}) \to \infty$ as $\| \boldsymbol{x} \| \to \infty$, then the equilibrium point is *globally asymptotically stable*, i.e. $\boldsymbol{x}(t) \to \boldsymbol{x}_e$ as $t \to \infty$ for any initial condition $\boldsymbol{x}(t_0)$.

Note that conditions 3 and 4 follow directly from LaSalle's invariance principle.

CHAPTER III

3 TRANSLATIONAL CONTROL

3.1 Translational Dynamics

In this section, we will describe the dynamics of a leader-follower pair shown in Figure 1. The development here follows that in [22]. Let $[X,Y,Z]$ frame denote an inertial frame. A commonly used inertial frame for earth orbits is the Geocentric Equatorial Frame, where X axis points in the vernal equinox direction, the XY plane is the earth's equatorial plane, and the Z axis coincides with the earth's axis of rotation and points northward. Let $\boldsymbol{F}_l$ and $\boldsymbol{F}_f$ denote the translational control forces for the leader and follower, respectively. Then, the dynamic equations for the translational motion of the leader and the follower in the $[X,Y,Z]$ frame respectively are given by

$$m_l\ddot{\boldsymbol{R}}+m_l\left(M+m_l\right)G\frac{\boldsymbol{R}}{\|\boldsymbol{R}\|^3}=\boldsymbol{F}_l \tag{3.1}$$

$$m_f\left(\ddot{\boldsymbol{R}}+\ddot{\boldsymbol{\rho}}\right)+m_f\left(M+m_f\right)G\left(\frac{\boldsymbol{R}+\boldsymbol{\rho}}{\|\boldsymbol{R}+\boldsymbol{\rho}\|^3}\right)=\boldsymbol{F}_f \tag{3.2}$$

where $\boldsymbol{R}$ is the inertial position of the leader satellite, $\boldsymbol{\rho}$ is the relative position of the follower spacecraft, m_l and m_f are the spacecraft masses, M is the earth's mass, and G is the universal gravitational constant. Disturbance forces due to solar radiation, aerodynamics, magnetic fields and higher order gravity terms are ignored here, although they can be easily included in the spacecraft dynamics assuming they are canceled by the spacecraft controls. Since $M>>m_i, i=l,f$, we can set $G\left(M+m_i\right)\approx GM$.

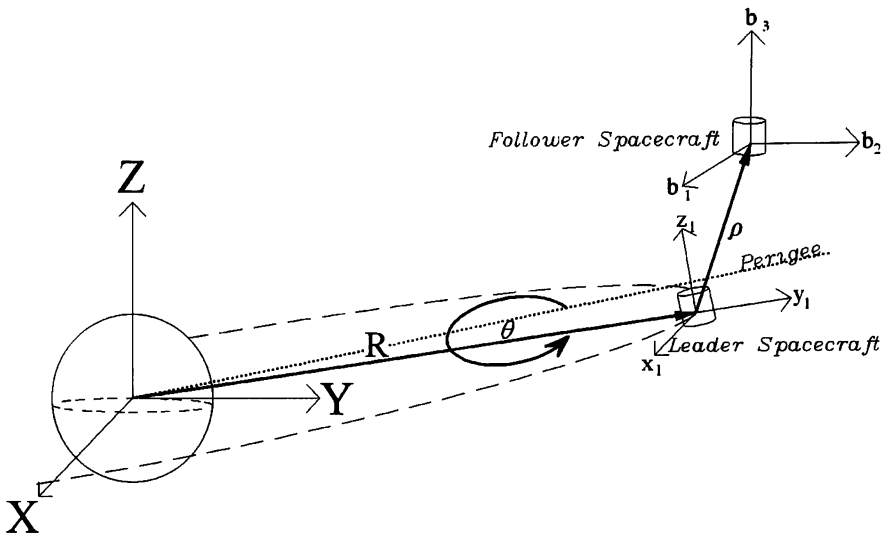


Figure 1 - Coordinate system for translational motion formulation

Rearranging (3.1), (3.2) and letting $\mu = GM$, the dynamic equation for the follower can be re-written as

$$m_f \ddot{\boldsymbol{\rho}} + m_f \mu \left(\frac{\boldsymbol{R}+\boldsymbol{\rho}}{\|\boldsymbol{R}+\boldsymbol{\rho}\|^3} - \frac{\boldsymbol{R}}{\|\boldsymbol{R}\|^3} \right) + \frac{m_f}{m_l} \boldsymbol{F}_l = \boldsymbol{F}_f \tag{3.3}$$

To write the dynamics (3.3) in terms of the leader coordinate frame $\left[x_l, y_l, z_l\right]$, we must obtain an expression for $\ddot{\rho}(t)$ in the leader coordinate frame. The desired equation of motion of the leader spacecraft is given by

$$\ddot{\boldsymbol{R}} + \mu \frac{\boldsymbol{R}}{\|\boldsymbol{R}\|^3} = 0 \tag{3.4}$$

Using a polar coordinate frame fixed at the center of earth as shown in Figure 1, the desired motion of the leader spacecraft can be described as

$$\ddot{r}_l - r_l\dot{\theta}^2 + \frac{\mu}{r_l^2} = 0 \tag{3.5}$$

$$r_l\ddot{\theta} + 2\dot{r}_l\dot{\theta} = 0 \tag{3.6}$$

where $r_l = \|\boldsymbol{R}\|$ is the orbital radius and θ is the true anomaly of the leader spacecraft. Applying orbital mechanics equations for elliptical orbits, the solution to these differential equations can be found as

$$r_l(t) = \frac{a_l\left(1-e_l^2\right)}{1+e_l\cos\theta(t)} \tag{3.7}$$

$$\dot{\theta}(t) = \frac{n\left(1+e_l\cos\theta(t)\right)^2}{\left(1-e_l^2\right)^{3/2}} \tag{3.8}$$

where a_l is the semi-major axis of the elliptical orbit of the leader spacecraft, e_l is the orbit eccentricity and n is the orbital rate defined as $n = \frac{2\pi}{T}$, where T is the orbital period. By differentiating (3.8) we get an expression for $\ddot{\theta}$:

$$\ddot{\theta}(t) = \frac{-2n^2e_l\left(1+e_l\cos\theta(t)\right)^3\sin\theta(t)}{\left(1-e_l^2\right)^3} \tag{3.9}$$

Noticing that the relative position vector $\boldsymbol{\rho}$ expressed in $[x_l, y_l, z_l]$ is given by

$$\boldsymbol{\rho} = x\hat{i}_l + y\hat{j}_l + z\hat{k}_l \tag{3.10}$$

and that the angular velocity of the leader coordinate frame is $\dot{\theta}\hat{k}_l$, the follower's relative acceleration as observed in the inertial frame can be derived:

$$\ddot{\boldsymbol{\rho}} = \left(\ddot{x} - 2\dot{\theta}\dot{y} - \dot{\theta}^2x - \ddot{\theta}y\right)\hat{i}_l + \left(\ddot{y} + 2\dot{\theta}\dot{x} - \dot{\theta}^2y + \ddot{\theta}x\right)\hat{j}_l + \ddot{z}\hat{k}_l \tag{3.11}$$

When this is substituted back into equation (3.3), the dynamic equation can then be written in the following form:

$$m_f\left(\ddot{\mathbf{p}}+C(\dot{\theta})\dot{\mathbf{p}}+N(\mathbf{p},\dot{\theta},\ddot{\theta},R\right)+\frac{m_f}{m_l}F_l=F_f \tag{3.12}$$

where

$$C(\dot{\theta})=2\dot{\theta}\begin{bmatrix}0 & -1 & 0\\ 1 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}, \tag{3.13}$$

$$N(\mathbf{p},\dot{\theta},\ddot{\theta},R)=\begin{bmatrix}\mu\dfrac{x}{\|R+\mathbf{p}\|^3}-\left(\dot{\theta}^2x+\ddot{\theta}y\right)\\ \mu\left(\dfrac{y+\|R\|}{\|R+\mathbf{p}\|^3}-\dfrac{1}{\|R\|^2}\right)-\left(\dot{\theta}^2y-\ddot{\theta}x\right)\\ \mu\dfrac{z}{\|R+\mathbf{p}\|^3}\end{bmatrix}, \tag{3.14}$$

and

$$\mathbf{p}(t)=\begin{bmatrix}x(t)\\ y(t)\\ z(t)\end{bmatrix},\ \dot{\mathbf{p}}(t)=\begin{bmatrix}\dot{x}(t)\\ \dot{y}(t)\\ \dot{z}(t)\end{bmatrix},\ \ddot{\mathbf{p}}(t)=\begin{bmatrix}\ddot{x}(t)\\ \ddot{y}(t)\\ \ddot{z}(t)\end{bmatrix}$$

Here $\mathbf{p}(t)$, $\dot{\mathbf{p}}(t)$, and $\ddot{\mathbf{p}}(t)$ are the relative position, relative velocity, and relative acceleration vectors as observed in the leader frame, respectively.

3.2 Translational Control Law

Let $\mathbf{p}_d=\left(x_d,y_d,z_d\right)^T$ denote a desired equilibrium relative position vector. The translational control problem is then to design a feedback control law F_f such that, starting from any initial relative position $\mathbf{p}(0)$ and relative velocity $\dot{\mathbf{p}}(0)$, the follower is driven to $\mathbf{p}=\mathbf{p}_d$ and $\dot{\mathbf{p}}_d=0$. Let $e(t)=\mathbf{p}(t)-\mathbf{p}_d$ denote the relative position error. Then

$$\dot{\boldsymbol{e}} = \dot{\mathbf{p}} \tag{3.15}$$

$$\ddot{\boldsymbol{e}} = \ddot{\mathbf{p}} \tag{3.16}$$

since $\mathbf{p}_d$ is a constant vector. Consequently, the equation of motion (3.12) for the follower becomes:

$$\ddot{\boldsymbol{e}} + \boldsymbol{C}(\dot{\theta})\dot{\boldsymbol{e}} + \boldsymbol{N}(\boldsymbol{e} + \mathbf{p}_d, \dot{\theta}, \ddot{\theta}, \boldsymbol{R}) + \frac{1}{m_l}\boldsymbol{F}_l = \frac{\boldsymbol{F}_f}{m_f} \tag{3.17}$$

Note that we will set $\boldsymbol{F}_l = 0$ assuming that the leader is on the desired orbit described by (3.5)-(3.6). Consider the following controller:

$$\boldsymbol{F}_f = m_f\left(\boldsymbol{C}(\dot{\theta})\dot{\boldsymbol{e}} + \boldsymbol{N}(\boldsymbol{e} + \mathbf{p}_d, \dot{\theta}, \ddot{\theta}, \boldsymbol{R})\right) - m_f\boldsymbol{B}\dot{\boldsymbol{e}} - m_f\boldsymbol{K}\boldsymbol{e} \tag{3.18}$$

where $\boldsymbol{B}$ and $\boldsymbol{K}$ are symmetric and positive definite matrices. The closed-loop dynamics for the system then becomes:

$$\ddot{\boldsymbol{e}} + \boldsymbol{B}\dot{\boldsymbol{e}} + \boldsymbol{K}\boldsymbol{e} = 0 \tag{3.19}$$

To prove that the control law (3.18) achieves the control objective, consider the following candidate Lyapunov function:

$$E = \frac{1}{2}\dot{\boldsymbol{e}}^T\dot{\boldsymbol{e}} + \frac{1}{2}\boldsymbol{e}^T\boldsymbol{K}\boldsymbol{e} \tag{3.20}$$

Taking the time derivative along the closed-loop trajectories yields

$$\dot{E} = \dot{\boldsymbol{e}}^T\ddot{\boldsymbol{e}} + \dot{\boldsymbol{e}}^T\boldsymbol{K}\boldsymbol{e} = -\dot{\boldsymbol{e}}^T\boldsymbol{B}\dot{\boldsymbol{e}} \tag{3.21}$$

Clearly, $\dot{E} \le 0$. Now it suffices to show that $\dot{E}$ is not identically zero along any solution of (3.19) other than the desired equilibrium $\boldsymbol{e} = 0$, $\dot{\boldsymbol{e}} = 0$. It is easily seen that $\dot{E} \equiv 0 \Rightarrow \dot{\boldsymbol{e}} \equiv 0 \Rightarrow \ddot{\boldsymbol{e}} \equiv 0$, which implies that $\boldsymbol{e} = 0$ as well, thus proving global asymptotic stability. This means that the proposed feedback control law drives the system to the

desired equilibrium from any $e(t_0)$, $\dot{e}(t_0)$. The feedback control law (3.18) can be written in terms of original variables as

$$\boldsymbol{F}_f = m_f\left(\boldsymbol{C}(\dot{\theta})\dot{\mathbf{p}} + \boldsymbol{N}(\mathbf{p},\dot{\theta},\ddot{\theta},\boldsymbol{R})\right) - m_f\boldsymbol{B}\dot{\mathbf{p}} - m_f\boldsymbol{K}(\mathbf{p}-\mathbf{p}_d) \tag{3.22}$$

When this is applied to the dynamics of the follower described in (3.12), the follower system should be driven to the desired position relative to the leader.

3.3 Matlab Results

The translational control described above was tested with a leader-follower system. Using Matlab and an ODE45 integrator, the control was applied in order to place the follower spacecraft at a particular location in the leader's frame.

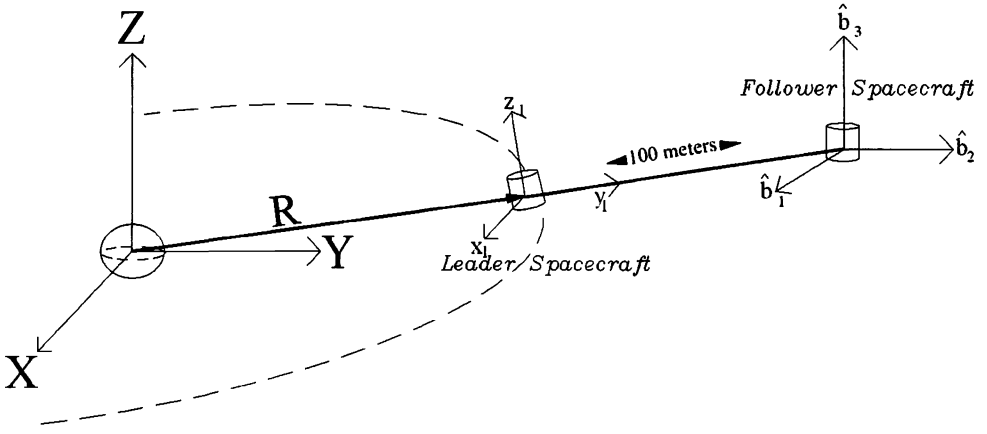


Figure 2 - Leader/Follower final location desired in Matlab simulation

As shown in Figure 2, the desired location in this example was chosen to be:

$$\mathbf{p}_d = \begin{bmatrix} 0 \\ 0.1 \\ 0 \end{bmatrix} km \tag{3.23}$$

Figure 3 shows the results of the simulation that corresponds to initial conditions

$$\mathbf{p}(0) = \begin{bmatrix} -0.1 & -0.5 & -0.2 \end{bmatrix}^T km \tag{3.24}$$

$$\dot{\mathbf{p}}(0) = \begin{bmatrix} -0.136 & 0 & 0.040 \end{bmatrix}^T km/h \tag{3.25}$$

Note that we set $m_f = 100\,kg$, $\mu = 5.166e12\,km^3/h^2$ and $e_l = 0$. The control gain matrices are

$$\boldsymbol{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \boldsymbol{K} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \tag{3.26}$$

Figure 3 displays the position of the follower in terms of $[x(t), y(t), z(t)]$ in the leader frame over time. It can be seen that $x \to 0$, $y \to 0.1\,km$, and $z \to 0$.

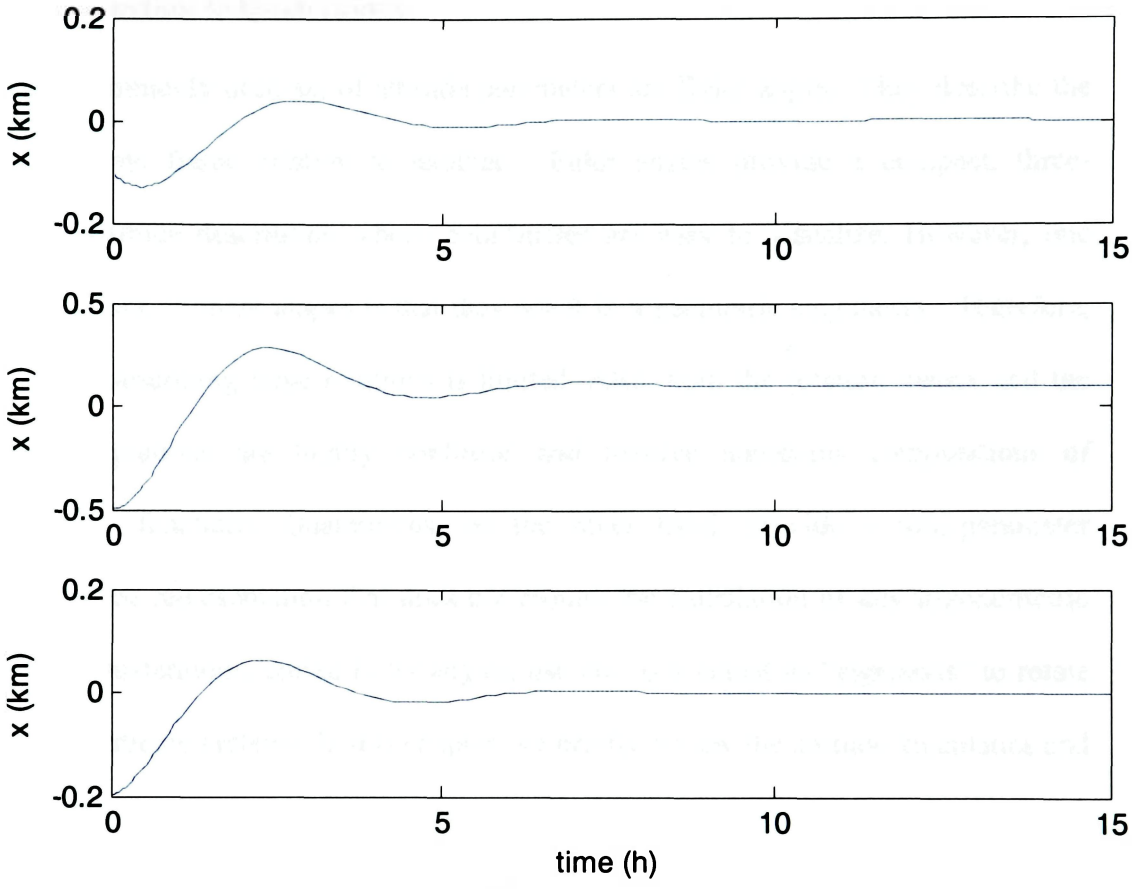


Figure 3 - Follower Spacecraft relative position

CHAPTER IV

4 ROTATIONAL CONTROL

4.1 Introduction to Quaternions

The most commonly used set of attitude parameters are Euler angles. They describe the attitude of one frame relative to another. Euler angles provide a compact, three-parameter attitude description whose coordinates are easy to visualize. However, one major drawback of these angles is that they result in a geometric singularity. Therefore, their use in describing large rotations is limited. Also, both the rotation matrix and the kinematic equations are highly nonlinear and involve numerous computations of trigonometric functions. Quaternions, on the other hand, provide a four-parameter singularity free representation that does not require the calculation of any trigonometric functions. Quaternions, unlike Euler angles, use one axis called an "eigenaxis" to rotate between coordinate systems. In this chapter, we briefly review the attitude kinematics and dynamics formulation used in this thesis to obtain the rotational equations of motion for a group of spacecraft. For full details, the reader is referred to [6] and [20].

4.2 Reference Frames and Rotations

Consider a right-handed orthonormal reference frame $\mathscr{F}_a$, whose three constituent vectors are $\hat{a}_1$, $\hat{a}_2$ and $\hat{a}_3$. Let $\cos\theta_1$, $\cos\theta_2$ and $\cos\theta_3$ be the direction cosines of a vector $\mathbf{r}$ as shown in Figure 4.

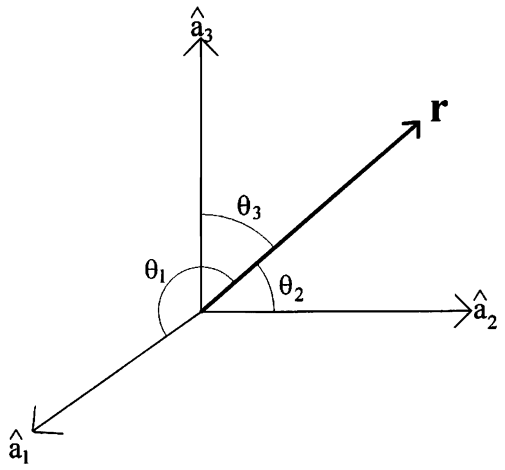


Figure 4 - Direction cosines between a vector r and a frame $\mathscr{F}_a$

Then, we write

$$\mathbf{r} = r\left(\hat{a}_1 \cos\theta_1 + \hat{a}_2 \cos\theta_2 + \hat{a}_3 \cos\theta_3\right) \tag{4.1}$$

where r is the length of $\mathbf{r}$. Now consider another right-handed orthonormal reference frame $\mathscr{F}_b$, with constituent vectors $\hat{b}_1$, $\hat{b}_2$ and $\hat{b}_3$. A relation between the two reference frames $\mathscr{F}_a$ and $\mathscr{F}_b$ can be written as:

$$\begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \end{bmatrix} \tag{4.2}$$

where c_{ij} is the direction cosine between $\hat{b}_i$ and $\hat{a}_j$. The matrix

$$\boldsymbol{C} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \tag{4.3}$$

is an orthonormal rotation matrix with the following properties:

$$\boldsymbol{C}\boldsymbol{C}^T = \boldsymbol{C}^T\boldsymbol{C} = \boldsymbol{I} \text{ and } \det(\boldsymbol{C}) = +1 \tag{4.4}$$

where I is the 3x3 identity matrix. The rotation matrix C relates components of a given vector $\mathbf{r}$ in the frames $\mathcal{F}_a$ and $\mathcal{F}_b$ as $\mathbf{r}_b = C\,\mathbf{r}_a$.

4.3 Rotational Kinematics

Euler's theorem states that the general rotation of a rigid body with one fixed point is a rotation about an axis through that point. Figure 5 illustrates the geometry pertaining to Euler's theorem.

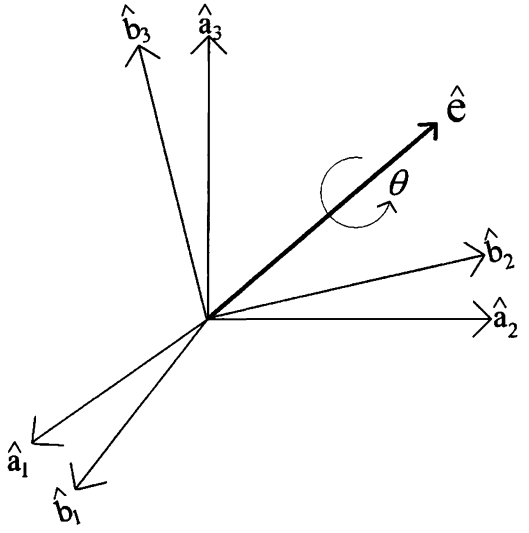


Figure 5 - Geometry pertaining to Euler's theorem

Now consider an arbitrary vector $\mathbf{r}$ as shown in Figure 6. As $\mathcal{F}_a$ rotates about an axis $\mathbf{e}$ (called an eigenaxis), by an angle θ (called an eigenangle), it will appear to an observer fixed in $\mathcal{F}_a$ that $\mathbf{r}$ is rotating about $\mathbf{e}$ through an angle $-\theta$; to this observer, the rotation corresponds to $\mathbf{r} \rightarrow \mathbf{r}'$, where

$$\mathbf{r}' = (\mathbf{e}\cdot\mathbf{r})\mathbf{e} - \mathbf{e}\times(\mathbf{e}\times\mathbf{r})\cos\theta - \mathbf{e}\times\mathbf{r}\sin\theta \tag{4.5}$$

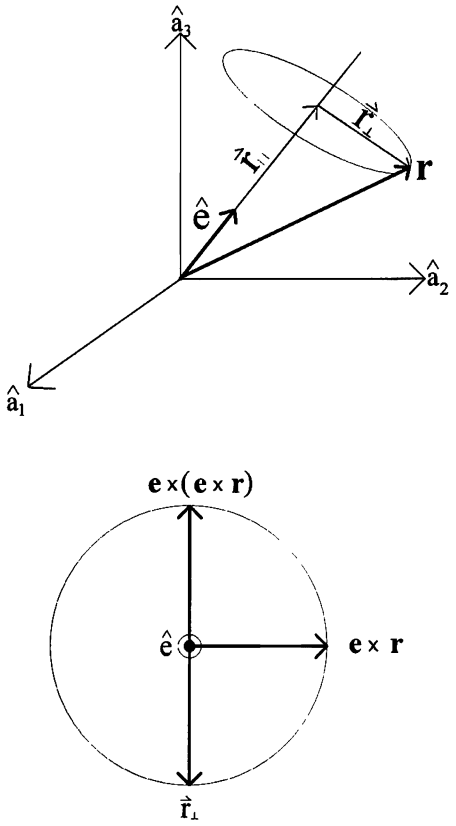


Figure 6 - Geometrical interpretation of the rotation matrix

Note that $\mathbf{e}^T\mathbf{e}=1$. The components of $\mathbf{r}'$ in $\mathcal{F}_b$ can then be written as

$$\mathbf{r}_b=\left[\mathbf{e}\mathbf{e}^T+\left(I-\mathbf{e}\mathbf{e}^T\right)\cos\theta-\boldsymbol{E}\sin\theta\right]\mathbf{r}_a \tag{4.6}$$

Thus, the rotation matrix can be expressed in terms of $\mathbf{e}$ and θ as

$$\boldsymbol{C}=\mathbf{e}\mathbf{e}^T+\left(I-\mathbf{e}\mathbf{e}^T\right)\cos\theta-\boldsymbol{E}\sin\theta \tag{4.7}$$

where $\boldsymbol{E}$ denotes the skew symmetric matrix satisfying $\mathbf{e}\times\mathbf{r}=\boldsymbol{E}\,\mathbf{r}$, which is given by

$$\boldsymbol{E}=\begin{bmatrix}0 & -e_3 & e_2\\ e_3 & 0 & -e_1\\ -e_2 & e_1 & 0\end{bmatrix} \tag{4.8}$$

In full matrix form, the rotation matrix becomes

$$\boldsymbol{C} = \begin{bmatrix} c\theta + e_1^2(1-c\theta) & e_1e_2(1-c\theta)+e_3s\theta & e_1e_3(1-c\theta)-e_2s\theta \\ e_2e_1(1-c\theta)-e_3s\theta & c\theta + e_2^2(1-c\theta) & e_2e_3(1-c\theta)+e_1s\theta \\ e_3e_2(1-c\theta)+e_2s\theta & e_3e_2(1-c\theta)-e_1s\theta & c\theta + e_3^2(1-c\theta) \end{bmatrix} \tag{4.9}$$

where $c\theta \triangleq \cos\theta$ and $s\theta \triangleq \sin\theta$.

Now quaternions (also called Euler parameters) can be defined as:

$$q_1 = e_1 \sin(\theta/2) \tag{4.10}$$

$$q_2 = e_2 \sin(\theta/2) \tag{4.11}$$

$$q_3 = e_3 \sin(\theta/2) \tag{4.12}$$

$$q_4 = \cos(\theta/2) \tag{4.13}$$

Like the eigenaxis vector $\mathbf{e} = (e_1, e_2, e_3)^T$, we define a vector $\mathbf{q} = (q_1, q_2, q_3)^T$ such that

$$\mathbf{q} = \mathbf{e}\sin(\theta/2) \tag{4.14}$$

Note that the quaternions are constrained by the following relationship:

$$\mathbf{q}^T\mathbf{q} + q_4^2 = q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \tag{4.15}$$

The rotation matrix can be parameterized in terms of quaternions as

$$\boldsymbol{C} = \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2+q_3q_4) & 2(q_1q_3-q_2q_4) \\ 2(q_2q_1-q_3q_4) & 1-2(q_3^2+q_1^2) & 2(q_2q_3+q_1q_4) \\ 2(q_3q_1+q_2q_4) & 2(q_3q_2-q_1q_4) & 1-2(q_1^2+q_2^2) \end{bmatrix} \tag{4.16}$$

Let $\boldsymbol{\omega}$ denote the angular velocity of $\mathscr{F}_b$ relative to $\mathscr{F}_a$. Then, as shown in [20], the kinematic differential equations for quaternions can be written as

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{4.17}$$

which can also be expressed as

$$\dot{\mathbf{q}} = \frac{1}{2}\left(q_4 \boldsymbol{\omega} - \boldsymbol{\omega} \times \mathbf{q}\right) \tag{4.18}$$

$$\dot{q}_4 = -\frac{1}{2}\boldsymbol{\omega}^T \mathbf{q} \tag{4.19}$$

4.4 Rotational Equations of Motion

We will specify the attitude of each spacecraft (the leader as well as the followers) relative to the inertial frame as opposed to using attitudes relative to the leader orbit frame as in [21]. Let $\mathscr{F}_a$ be the inertial reference frame $[X, Y, Z]$ and let $\mathscr{F}_{bi}$ denote the ith spacecraft body-fixed frame ($i = l$ corresponds to the leader and $i = 1, 2, 3, \cdots$ correspond to the followers) as shown in Figure 7. The orientation of $\mathscr{F}_{bi}$ relative to $\mathscr{F}_a$ is given by the quaternion

$$\mathbf{q}_i = \mathbf{e}_i \sin\left(\theta_i / 2\right) \tag{4.20}$$

$$q_{i4} = \cos\left(\frac{\theta_i}{2}\right) \tag{4.21}$$

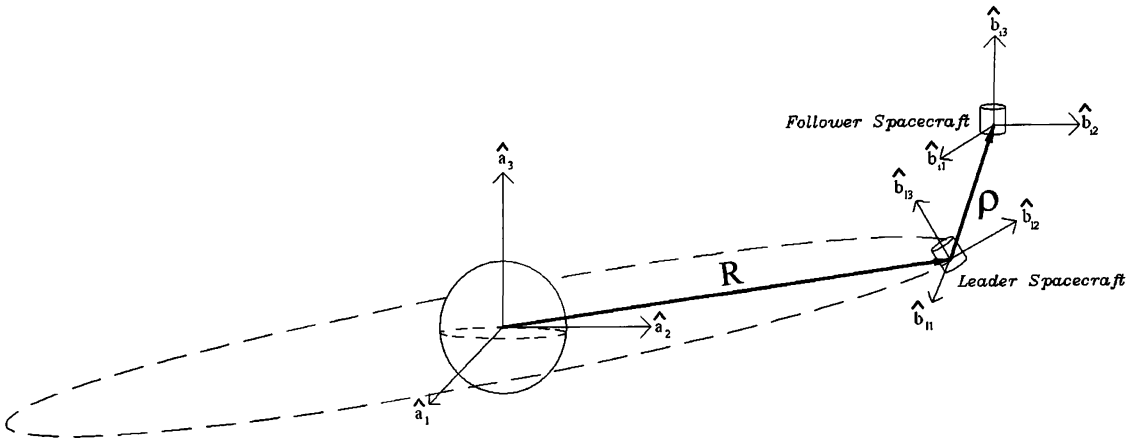


Figure 7 - Coordinate system used for rotational motion formulation

Let

$$\omega_i = \begin{pmatrix} \omega_{i1} \\ \omega_{i2} \\ \omega_{i3} \end{pmatrix} \tag{4.22}$$

denote the angular velocity of the ith spacecraft relative to the inertial frame $\mathcal{F}_a$. Then the kinematics for the ith spacecraft can be written as:

$$\dot{\mathbf{q}}_i = \frac{1}{2}\left(q_{i4}\boldsymbol{\omega}_i - \boldsymbol{\omega}_i \times \mathbf{q}_i\right) \tag{4.23}$$

$$\dot{q}_{i4} = -\frac{1}{2}\boldsymbol{\omega}_i^T \mathbf{q}_i \tag{4.24}$$

Let $\boldsymbol{\tau}_i$ denote the control torque vector for the ith spacecraft. Then the rotational dynamics can be written as

$$\boldsymbol{J}_i\dot{\boldsymbol{\omega}}_i + \tilde{\boldsymbol{\omega}}_i\boldsymbol{J}_i\boldsymbol{\omega}_i = \boldsymbol{\tau}_i \tag{4.25}$$

where $\boldsymbol{J}_i$ is the inertia matrix for the ith spacecraft, which is given by

$$\boldsymbol{J}_i = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix} \tag{4.26}$$

and $\tilde{\omega}_i$ is the skew-symmetric matrix formed from ω_i:

$$\tilde{\omega}_i = \begin{bmatrix} 0 & -\omega_{i3} & \omega_{i2} \\ \omega_{i3} & 0 & -\omega_{i1} \\ -\omega_{i2} & \omega_{i1} & 0 \end{bmatrix} \tag{4.27}$$

Clearly, $\tilde{\omega}_i \boldsymbol{J}_i \omega_i = \omega_i \times \boldsymbol{J}_i \omega_i$ and, thus, both notations can be used interchangeably.

4.5 Rotational Control Law

The equations of motion for the *i*th spacecraft are given by (4.23)-(4.25). Now let the desired attitude trajectory for the *i*th spacecraft be specified by the desired quaternions

$$\left(q_{i1d}(t), q_{i2d}(t), q_{i3d}(t), q_{i4d}(t)\right) = \left(\mathbf{q}_{id}^T(t), q_{i4d}(t)\right)$$

and the desired angular velocity $\omega_{id}(t)$ relative to $\mathcal{F}_a$. The desired quaternions for the *i*th spacecraft then satisfy

$$\dot{\mathbf{q}}_{id} = \frac{1}{2}\left(q_{i4d}\boldsymbol{\omega}_{id} - \boldsymbol{\omega}_{id} \times \mathbf{q}_{id}\right) \tag{4.28}$$

$$\dot{q}_{i4d} = -\frac{1}{2}\boldsymbol{\omega}_{id}{}^T \mathbf{q}_{id} \tag{4.29}$$

The attitude error quaternions for the *i*th spacecraft are computed using the desired quaternions $\left(q_{i1d}, q_{i2d}, q_{i3d}, q_{i4d}\right)$ and the current attitude quaternions $\left(q_{i1}, q_{i2}, q_{i3}, q_{i4}\right)$ as follows:

$$\begin{bmatrix} q_{i1e} \\ q_{i2e} \\ q_{i3e} \\ q_{i4e} \end{bmatrix} = \begin{bmatrix} q_{i4d} & q_{i3d} & -q_{i2d} & -q_{i1d} \\ -q_{i3d} & q_{i4d} & q_{i1d} & -q_{i2d} \\ q_{i2d} & -q_{i1d} & q_{i4d} & -q_{i3d} \\ q_{i1d} & q_{i2d} & q_{i3d} & q_{i4d} \end{bmatrix} \begin{bmatrix} q_{i1} \\ q_{i2} \\ q_{i3} \\ q_{i4} \end{bmatrix} \tag{4.30}$$

Let $\omega_{ie} = \omega_i - \omega_{id}$ denote the angular velocity error. Then the differential equations for the attitude error quaternions are given by

$$\dot{\mathbf{q}}_{ie} = \frac{1}{2}\left(q_{i4e}\omega_{ie} - \omega_{ie} \times \mathbf{q}_{ie}\right) + \mathbf{q}_{ie} \times \omega_{ie} \tag{4.31}$$

$$\dot{q}_{i4e} = -\frac{1}{2}\omega_{ie}{}^{T}\mathbf{q}_{ie} \tag{4.32}$$

We will assume that ω_{id} is constant so that $\dot{\omega}_{ie} = \dot{\omega}_i$. Clearly, the dynamic equations given by (4.25) can then be rewritten as

$$J_i\dot{\omega}_{ie} + (\omega_{ie} + \omega_{id}) \times J_i(\omega_{ie} + \omega_{id}) = \tau_i \tag{4.33}$$

The goal now is to design a feedback control τ_i for the spacecraft to achieve the desired attitude and the desired angular velocity.

Consider the following controller:

$$\tau_i = -k_i J_i \mathbf{q}_{ie} - c_i J_i \omega_{ie} + \omega_i \times J_i \omega_i \tag{4.34}$$

where k_i and c_i are positive control parameters. The closed-loop dynamics can be written as

$$\dot{\omega}_{ie} = -k_i \mathbf{q}_{ie} - c_i \omega_{ie} \tag{4.35}$$

$$\dot{\mathbf{q}}_{ie} = \frac{1}{2}\left(q_{i4e}\omega_{ie} - \omega_{ie} \times \mathbf{q}_{ie}\right) + \mathbf{q}_{ie} \times \omega_{ie} \tag{4.36}$$

$$\dot{q}_{i4e} = -\frac{1}{2}\omega_{ie}{}^{T}\mathbf{q}_{ie} \tag{4.37}$$

To prove that the control law (4.34) achieves the control objective, consider the following candidate Lyapunov function:

$$E_i = \frac{1}{2k_i}\boldsymbol{\omega}_{ie}{}^T\boldsymbol{\omega}_{ie} + \mathbf{q}_{ie}{}^T\mathbf{q}_{ie} + \left(q_{i4e} - 1\right)^2 \tag{4.38}$$

The time derivative of E_i along the trajectories of this closed loop system (4.35)-(4.37) can be computed as

$$\dot{E}_i = \frac{\boldsymbol{\omega}_{ie}{}^T\dot{\boldsymbol{\omega}}_{ie}}{k_i} + 2\mathbf{q}_{ie}{}^T\dot{\mathbf{q}}_{ie} + 2\left(q_{i4e} - 1\right)\dot{q}_{i4e} \tag{4.39}$$

which simplifies to

$$\dot{E}_i = -\frac{c_i}{k_i}\boldsymbol{\omega}_{ie}{}^T\boldsymbol{\omega}_{ie} \leq 0 \tag{4.40}$$

Now it suffices to show that $\dot{E}_i$ is not identically zero along any solution of (4.35)-(4.37) other than the desired equilibrium $\boldsymbol{\omega}_{ie} = 0, \mathbf{q}_{ie} = 0$, and $q_{i4e} = 1$. It can be easily seen that $\dot{E}_i \equiv 0 \Rightarrow \boldsymbol{\omega}_{ie} \equiv 0 \Rightarrow \dot{\boldsymbol{\omega}}_{ie} = 0$, which implies $\mathbf{q}_{ie} = 0$ and $q_{i4e} = 1$ as well, thus proving global asymptotic stability. This means that the proposed control law achieves the objective.

4.6 Matlab Results

To test the effectiveness of the previously discussed control scheme, Matlab was used to simulate the controller response. The Matlab program applied the control mentioned in the previous section using ODE45 to show that the desired quaternion vector can be reached for a system of two satellites. In the following example, the desired quaternion was:

$$\begin{bmatrix} \mathbf{q}_{id} \\ q_{i4} \end{bmatrix} = \begin{bmatrix} 0 \\ 0.50 \\ 0 \\ 0.866 \end{bmatrix} \tag{4.41}$$

Figures 8-9 display the elements of the quaternion vectors for the leader and the follower spacecraft over a time of 100 seconds.

Clearly the desired quaternion vector was achieved in a short time period for both spacecraft in this example. To be exact, the quaternion reached for both spacecraft was:

$$\begin{bmatrix} \mathbf{q}_{id} \\ q_{i4} \end{bmatrix} = \begin{bmatrix} 0 \\ 0.4996 \\ 0 \\ 0.8654 \end{bmatrix} \tag{4.42}$$

This accuracy and efficiency may be further improved through the use of other Matlab integrators, reducing the number of computations, and simplifying loop structures.

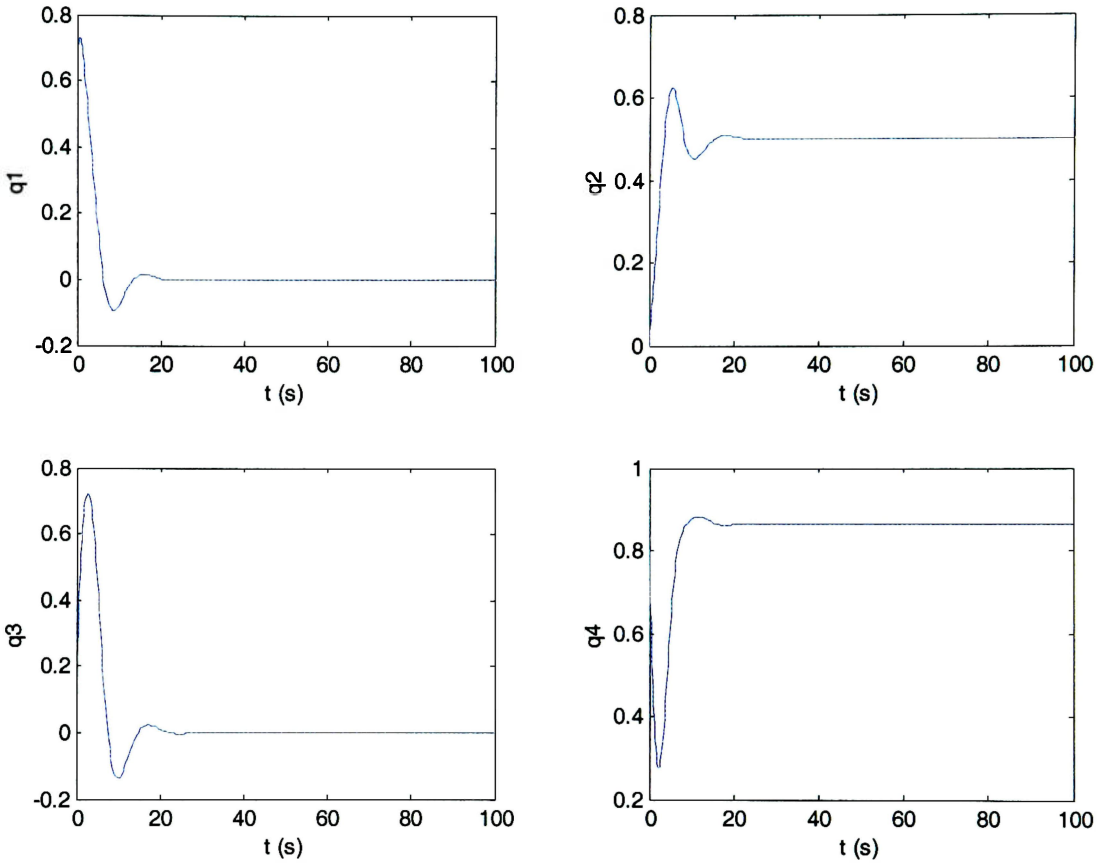


Figure 8 - Quaternions for the Leader Spacecraft

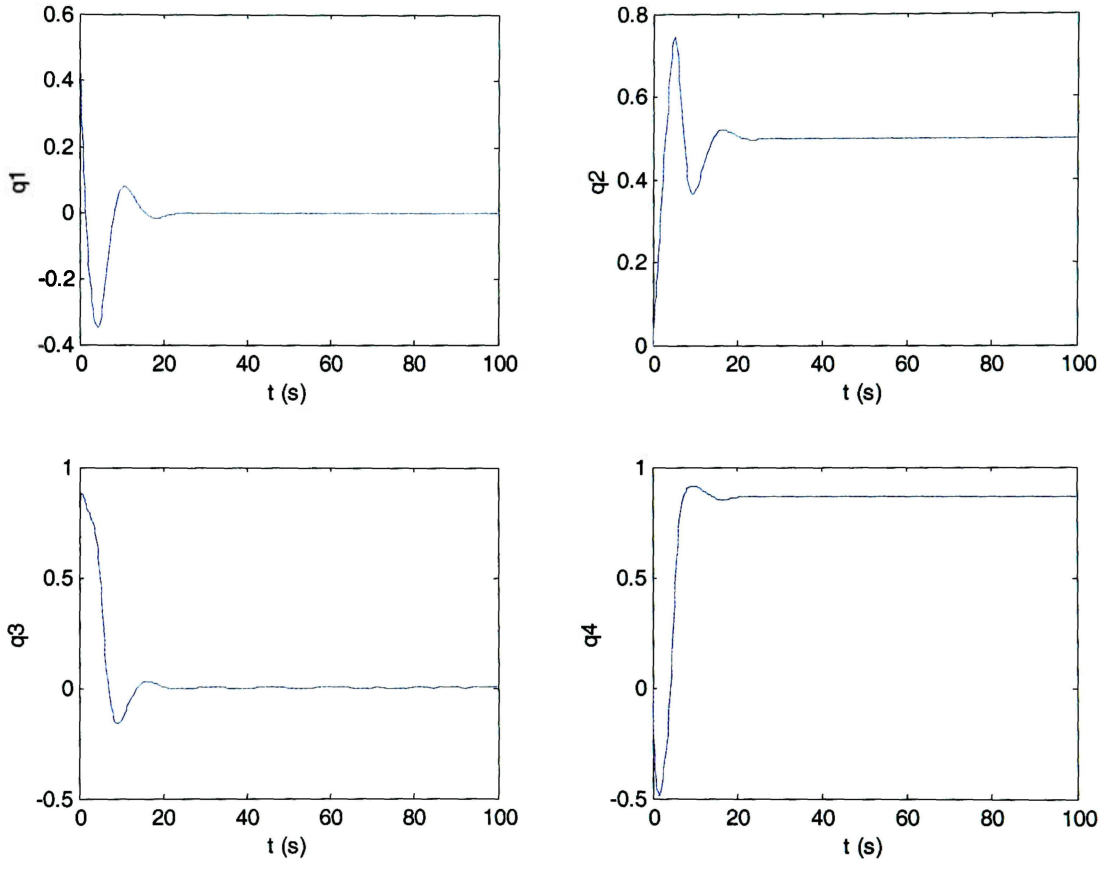


Figure 9 - Quaternions for the Follower Spacecraft

CHAPTER V

5 CONTROL SIMULATION

5.1 Matlab Simulation

Due to the fact that the translational and rotational dynamics are not coupled, the incorporation of the two into a single program is relatively simple. The combined dynamics and control program in APPENDIX A simulates both control schemes through the use of a Fourth-Order Runge-Kutta integration program within two separate control loops. Each loop cycles through the appropriate number of spacecraft depending on the size of the formation and simulates the controlled motion. After generating results regarding the spacecraft's attitude and position and converting to the appropriate units, the Matlab program generates simple text files that can be loaded into Satellite Tool Kit. The spacecraft system in this simulation consists of seven spacecraft: one leader and six followers. In this simulation, the six follower spacecraft form a circular pattern around the leader spacecraft in order to mimic a potential space-based interferometer or telescope. The spacing between the spacecraft is 0.5 kilometers, allowing the group to take the form of a hexagon consisting of six equilateral triangles. This arrangement, shown in Figure 10, will be achieved starting from any arbitrary initial states of the spacecraft.

In order to produce a clear view of both translational and rotational control in STK, the rotational control simulation begins after the spacecraft have come close to reaching their desired circular formation. Also, in original Matlab simulations (See Chapter 4), the final quaternion arrangement did not attempt to match the orbital rate. In this final simulation,

the orbital rate is matched in order for the formation to continue to face a specific direction as a group.

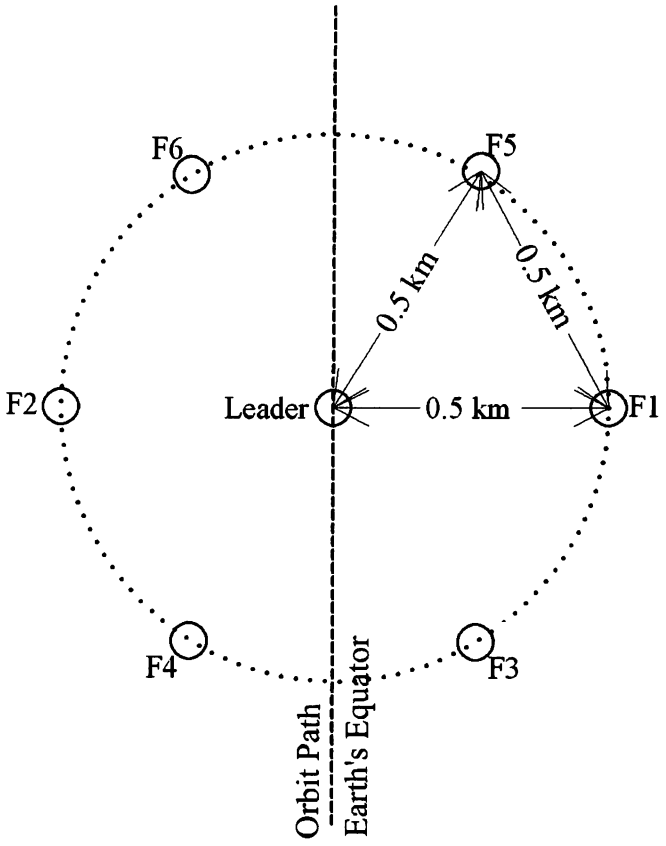


Figure 10 - Final desired seven spacecraft formation pattern

5.1.1 Translational Control

In preparation of a 3D animation using STK, the control law described in (3.18) was again used to simulate the formation displayed in Figure 10. The initial positions and velocities used in this formation as well as in subsequent reformation programs were chosen as follows:

$$
\begin{aligned}
\mathbf{p}_1(0) &= [-0.5 \quad 0.8 \quad -0.1]^T \, km & \dot{\mathbf{p}}_1(0) &= [-0.136 \quad 0 \quad 0.040]^T \, km/h \\
\mathbf{p}_2(0) &= [0.7 \quad 0.5 \quad 0.2]^T \, km & \dot{\mathbf{p}}_2(0) &= [-0.136 \quad 0 \quad 0.040]^T \, km/h \\
\mathbf{p}_3(0) &= [0.2 \quad 1.0 \quad 1.0]^T \, km & \dot{\mathbf{p}}_3(0) &= [-0.136 \quad 0 \quad 0.040]^T \, km/h \\
\mathbf{p}_4(0) &= [0.6 \quad 1.0 \quad -1.0]^T \, km & \dot{\mathbf{p}}_4(0) &= [-0.136 \quad 0 \quad 0.040]^T \, km/h \\
\mathbf{p}_5(0) &= [0.4 \quad 0.8 \quad 1.0]^T \, km & \dot{\mathbf{p}}_5(0) &= [-0.136 \quad 0 \quad 0.040]^T \, km/h \\
\mathbf{p}_6(0) &= [0.5 \quad 0.2 \quad 0.25]^T \, km & \dot{\mathbf{p}}_6(0) &= [-0.136 \quad 0 \quad 0.040]^T \, km/h
\end{aligned}
\tag{5.1}
$$

The desired positions of the six follower spacecraft in the leader frame corresponding to the formation in Figure 10 are given as

$$
\begin{aligned}
\mathbf{p}_{d1} &= (x_d, y_d, z_d)^T = (0.50 \quad 0 \quad 0)^T \, km \\
\mathbf{p}_{d2} &= (x_d, y_d, z_d)^T = (-0.50 \quad 0 \quad 0)^T \, km \\
\mathbf{p}_{d3} &= (x_d, y_d, z_d)^T = (0.25 \quad 0 \quad -0.433)^T \, km \\
\mathbf{p}_{d4} &= (x_d, y_d, z_d)^T = (-0.25 \quad 0 \quad 0.433)^T \, km \\
\mathbf{p}_{d5} &= (x_d, y_d, z_d)^T = (0.25 \quad 0 \quad 0.433)^T \, km \\
\mathbf{p}_{d6} &= (x_d, y_d, z_d)^T = (-0.25 \quad 0 \quad -0.433)^T \, km
\end{aligned}
\tag{5.2}
$$

As discussed earlier in Chapter III, the leader is assumed to be in a circular orbit. The Matlab results displaying the positions of all seven spacecraft over 20 hours can be seen in Figures 11-16. Clearly, these plots show that the formation described in Figure 10 has been attained.

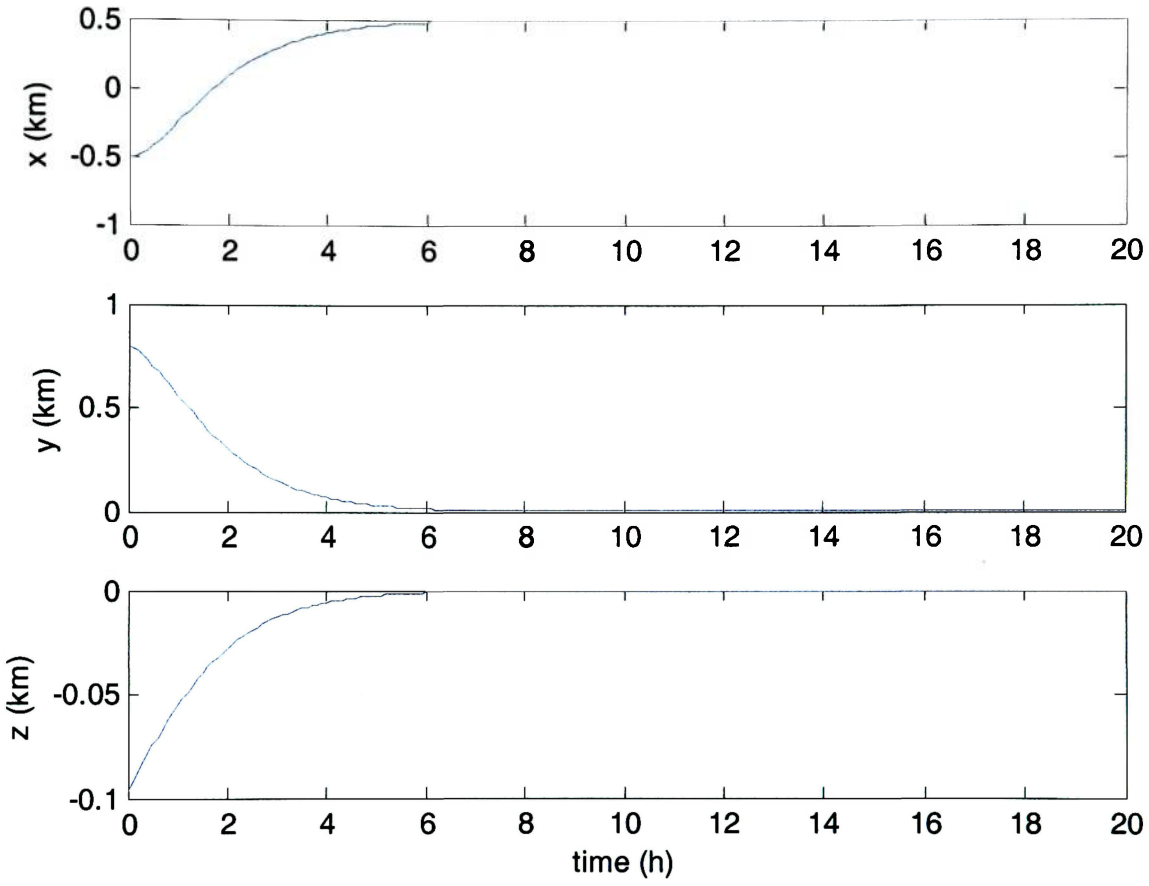


Figure 11 - Position of Follower Spacecraft 1 over time

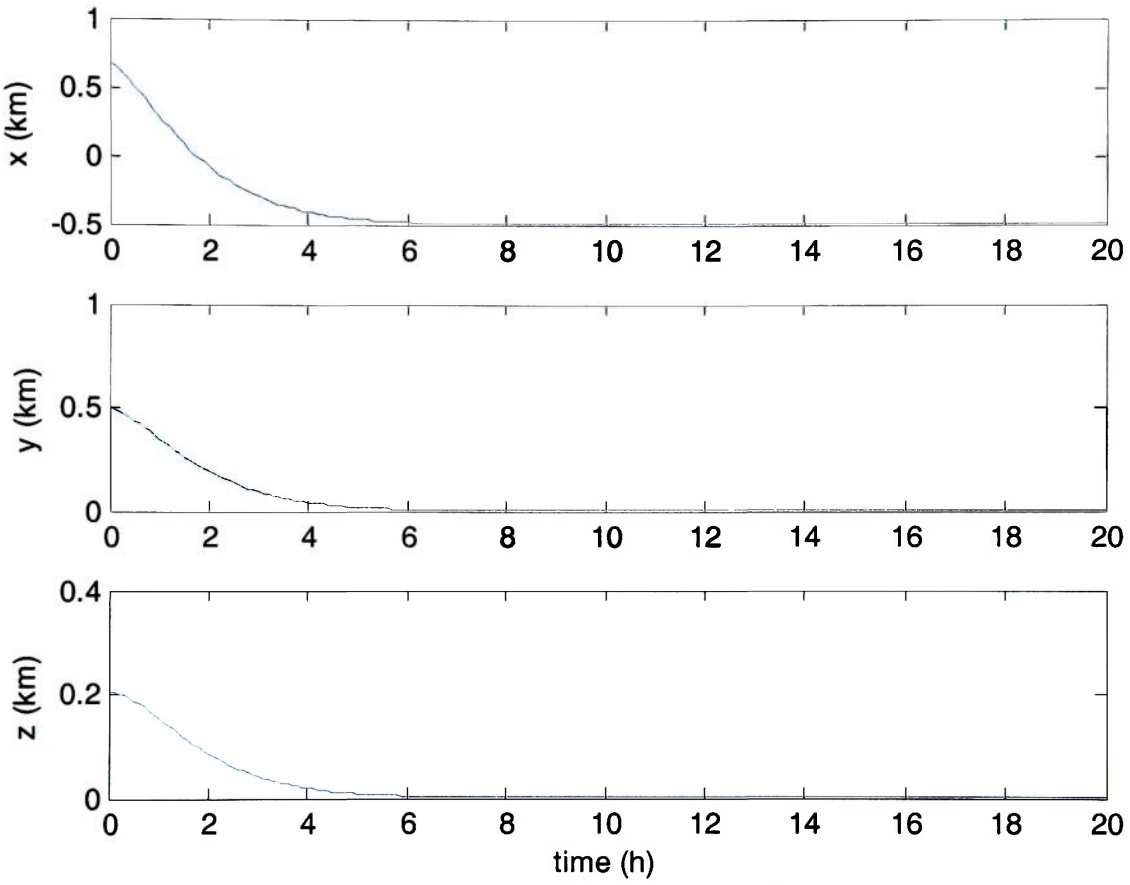


Figure 12 - Position of Follower Spacecraft 2 over time

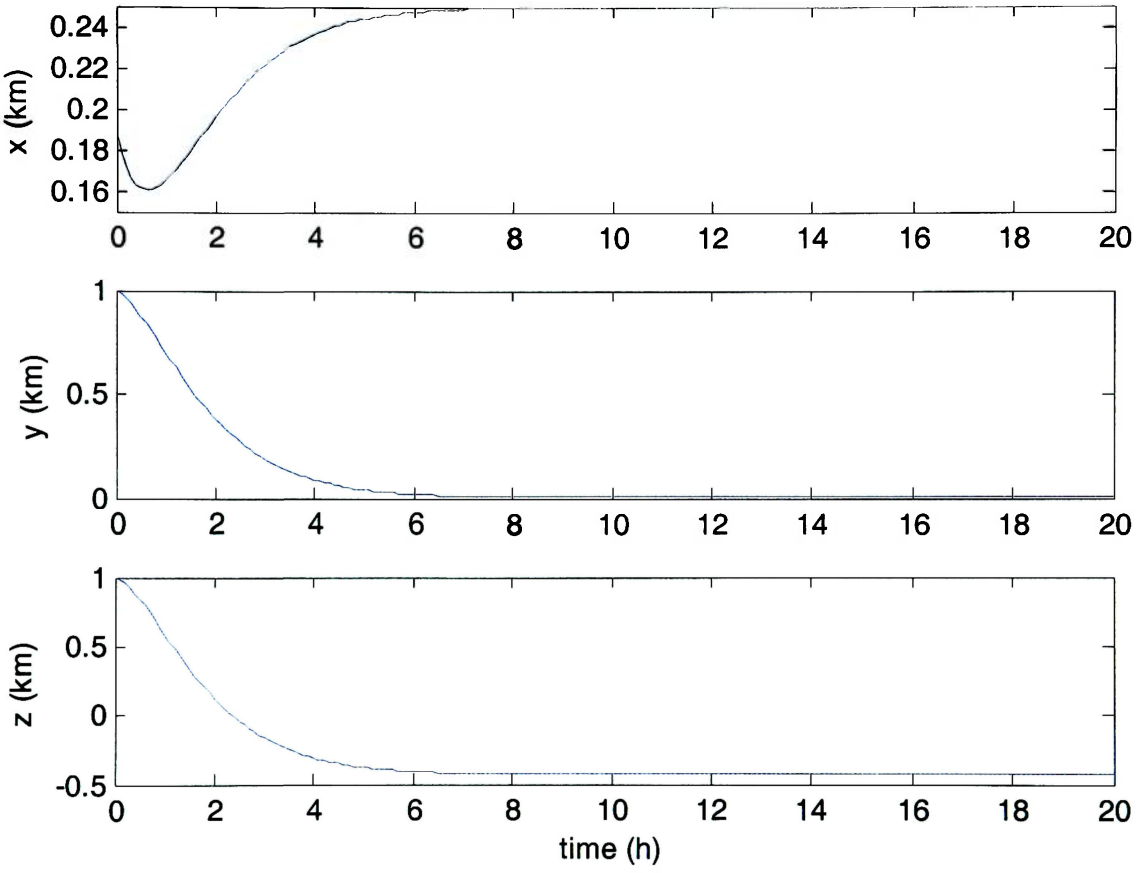


Figure 13 - Position of Follower Spacecraft 3 over time

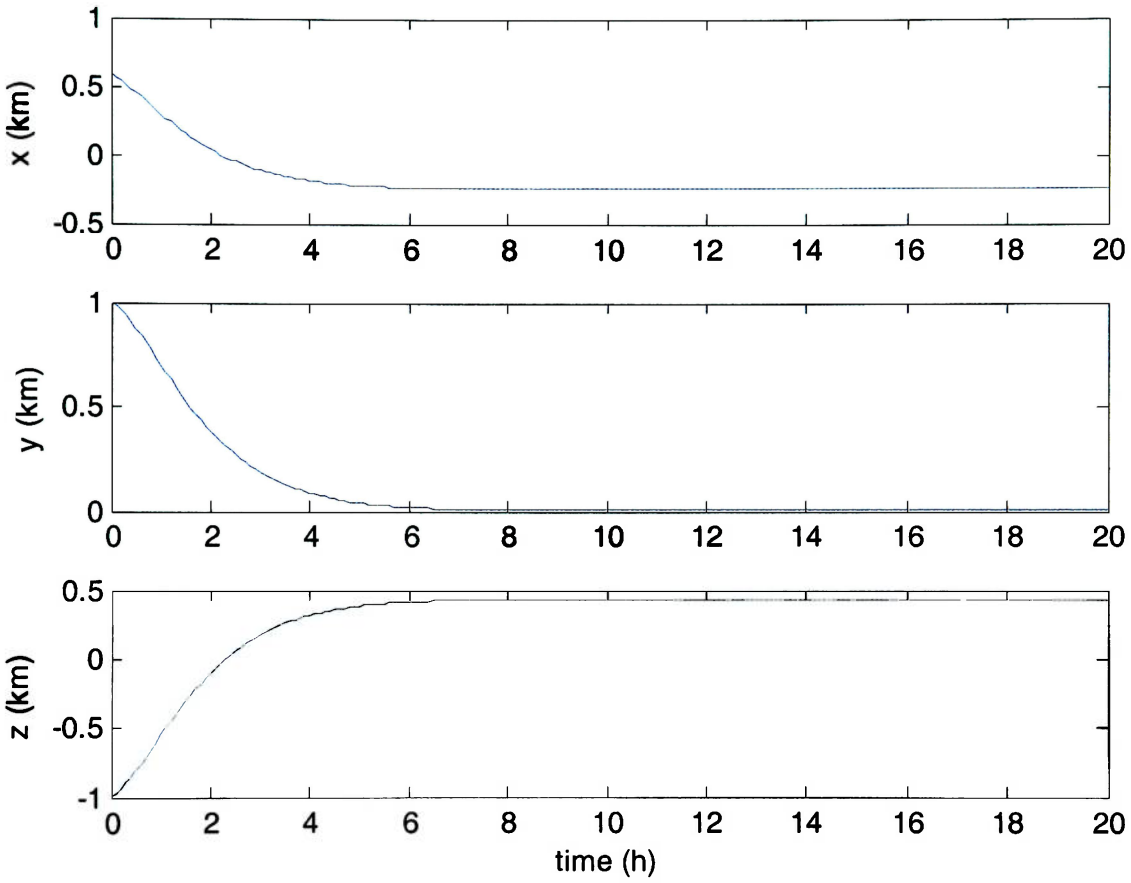


Figure 14 - Position of Follower Spacecraft 4 over time

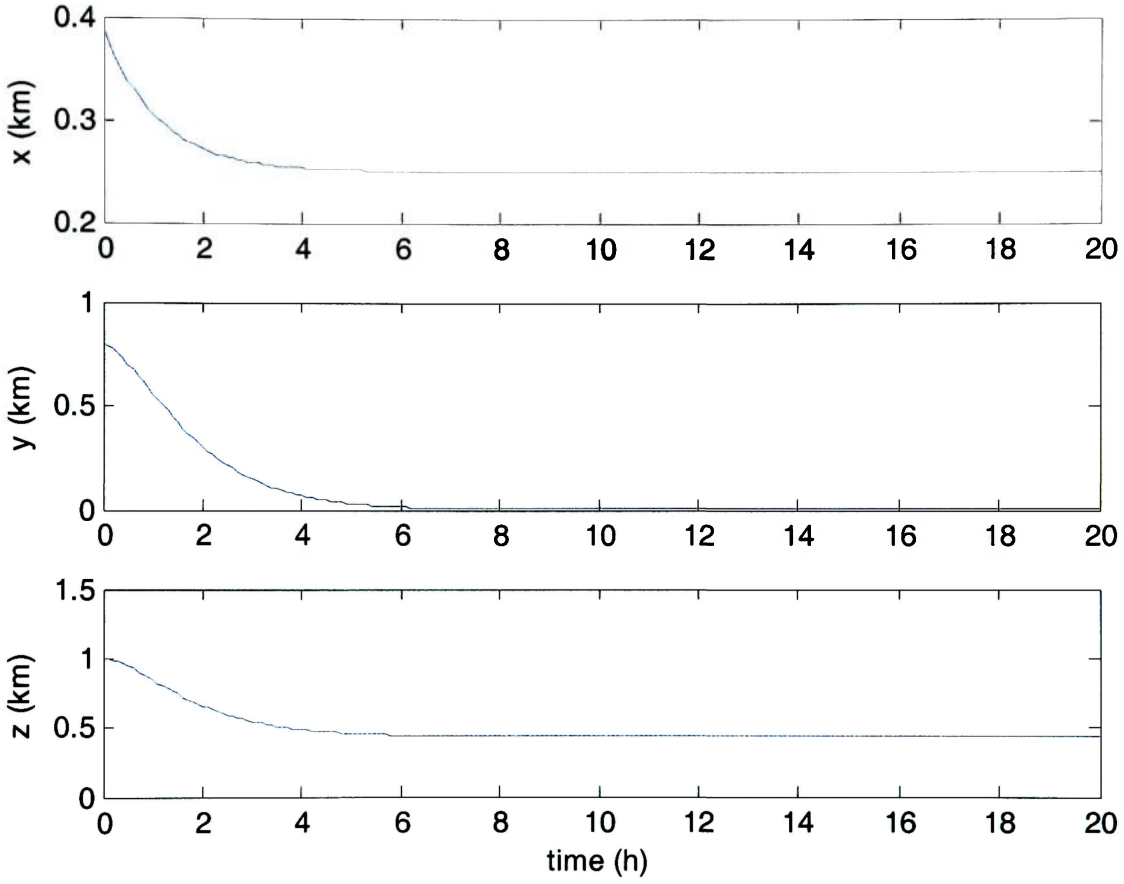


Figure 15 - Position of Follower Spacecraft 5 over time

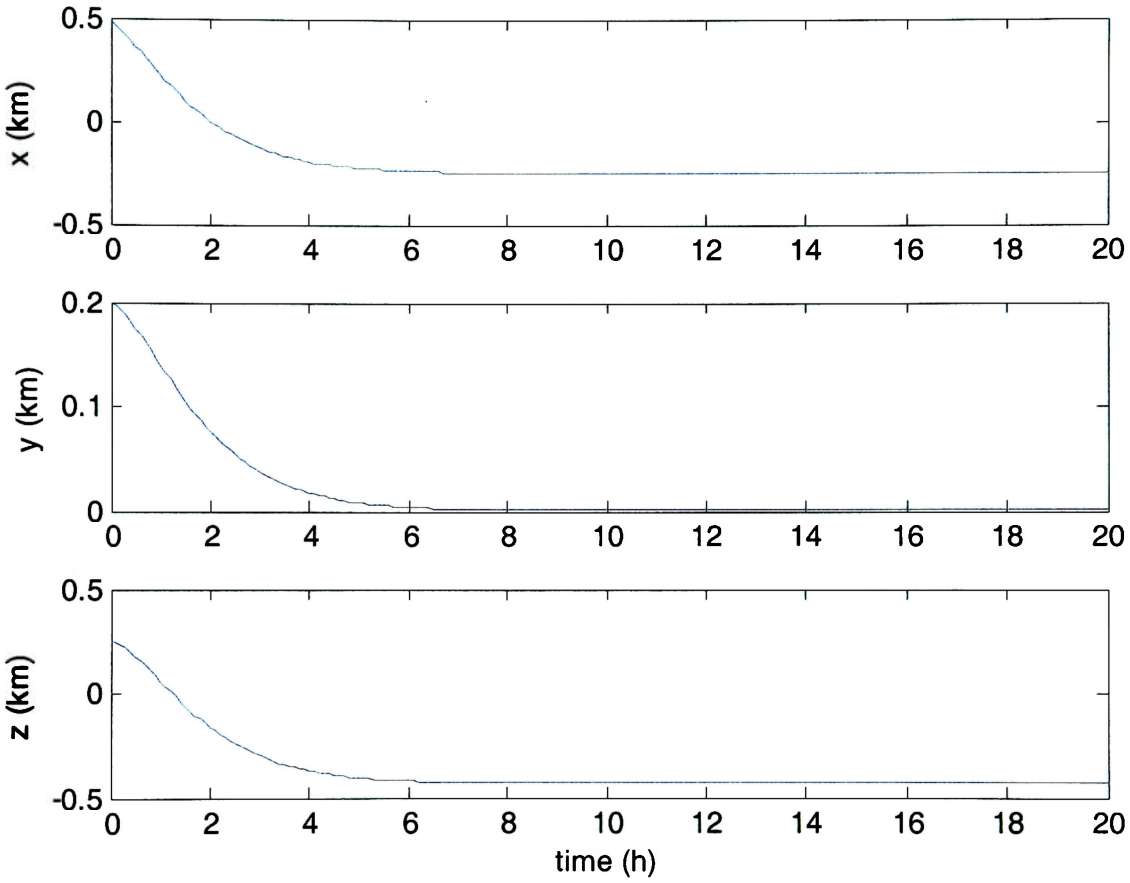


Figure 16 - Position of Follower Spacecraft 6 over time

5.1.2 Rotational Control

Considering some of the many uses of satellite formations, it is likely that the formation will require a consistent attitude adjustment to match it's "target" in some way. For example, if the function of the formation involved targeting an object on the Earth's surface or a region in the upper atmosphere, the formation would need to rotate at a consistent rate to compensate. In other words, the desired quaternion arrangement would be dynamic, adjusting according to the satellites' orbital rates. This issue was disregarded in earlier chapters, but added here for simulation purposes.

Many types of targeting could apply to this system, however, only Earth targeting is considered in this thesis. As stated earlier, the desired quaternions must be dynamic, allowing the spacecraft to rotate at their orbital rate. To achieve this, the desired angular velocities are chosen as

$$\omega_{id} = \begin{bmatrix} \omega_{i1d} \\ \omega_{i2d} \\ \omega_{i3d} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ n \end{bmatrix} \tag{5.3}$$

where n is the orbital rate For the leader's circular orbit, the orbital rate is calculated for as follows:

$$n = \frac{\sqrt{\mu}}{R^{3/2}} = 3.176\ rad/h \tag{5.4}$$

where $R = 8{,}000\ km$ (radius of the orbit) and $\mu = 5.166\ km^3/h^2$ (Earth's gravitational parameter).

Clearly, in this case, equations (4.28)-(4.29) for the desired quaternions can be written as

$$\begin{bmatrix} \dot{q}_{i1d} \\ \dot{q}_{i2d} \\ \dot{q}_{i3d} \\ \dot{q}_{i4d} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & n & 0 & 0 \\ -n & 0 & 0 & 0 \\ 0 & 0 & 0 & n \\ 0 & 0 & -n & 0 \end{bmatrix} \begin{bmatrix} q_{i1d} \\ q_{i2d} \\ q_{i3d} \\ q_{i4d} \end{bmatrix} \tag{5.5}$$

Solving these equations with the initial desired quaternions $\mathbf{q}_{id}(0) = 0$ and $q_{i4d}(0) = 1$ yields the desired quaternion trajectory

$$\begin{bmatrix} q_{1id} \\ q_{2id} \\ q_{3id} \\ q_{4id} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sin(nt/2) \\ \cos(nt/2) \end{bmatrix} \tag{5.6}$$

We will choose the $\mathcal{F}_{bi}$ frame to be the principal axes frame for the ith spacecraft with an inertia matrix of the form

$$\boldsymbol{J}_i = \begin{bmatrix} J_{i1} & 0 & 0 \\ 0 & J_{i2} & 0 \\ 0 & 0 & J_{i3} \end{bmatrix} \tag{5.7}$$

Thus, the rotational control law for the ith spacecraft can be written as

$$\boldsymbol{\tau}_i = -k_i \begin{bmatrix} J_{i1} q_{i1e} \\ J_{i2} q_{i2e} \\ J_{i3} q_{i3e} \end{bmatrix} - c_i \begin{bmatrix} J_{i1} \omega_{i1} \\ J_{i2} \omega_{i2} \\ J_{i3} \omega_{i3} \end{bmatrix} - \begin{bmatrix} (J_{i2} - J_{i3}) \omega_{i2} \omega_{i3} \\ (J_{i3} - J_{i2}) \omega_{i3} \omega_{i1} \\ (J_{i1} - J_{i2}) \omega_{i1} \omega_{i2} \end{bmatrix} \tag{5.8}$$

The principal inertias for all the spacecraft are identical and given by

$$\begin{aligned} J_{i1} &= 33\ kg \cdot m^2 \\ J_{i2} &= 33\ kg \cdot m^2 \\ J_{i3} &= 50\ kg \cdot m^2 \end{aligned} \tag{5.9}$$

In the simulations, the initial angular velocities are given by

$$\omega_{i1}(0) = \omega_{i2}(0) = \omega_{i3}(0) = 2.78e-4\ rad/h \tag{5.10}$$

The initial quaternions for each spacecraft is given as follows:

$$\begin{aligned}
&\textit{Leader} : (q_{l1}(0)\ \ q_{l2}(0)\ \ q_{l3}(0)\ \ q_{l4}(1)) = (0\ \ 0\ \ 0\ \ 1)\\
&\textit{Follower}\,1: (q_{11}(0)\ \ q_{12}(0)\ \ q_{13}(0)\ \ q_{14}(0)) = (0.6\ \ 0\ \ 0.8\ \ 0)\\
&\textit{Follower}\,2: (q_{21}(0)\ \ q_{22}(0)\ \ q_{23}(0)\ \ q_{24}(0)) = (0.6\ \ 0.8\ \ 0\ \ 0)\\
&\textit{Follower}\,3: (q_{31}(0)\ \ q_{32}(0)\ \ q_{33}(0)\ \ q_{34}(0)) = (0.8\ \ 0\ \ 0.6\ \ 0)\\
&\textit{Follower}\,4: (q_{41}(0)\ \ q_{42}(0)\ \ q_{43}(0)\ \ q_{44}(0)) = (0.8\ \ 0.6\ \ 0\ \ 0)\\
&\textit{Follower}\,5: (q_{51}(0)\ \ q_{52}(0)\ \ q_{53}(0)\ \ q_{54}(0)) = (0\ \ 0\ \ 0.8\ \ 0.6)\\
&\textit{Follower}\,6: (q_{61}(0)\ \ q_{62}(0)\ \ q_{63}(0)\ \ q_{64}(0)) = (0\ \ 0\ \ 0.6\ \ 0.8)
\end{aligned} \tag{5.11}$$

The control parameters are chosen as $c_i = k_i = 10$ for each spacecraft. The Matlab results using these parameters over a time period of 20 hours can be seen in Figures 17-23. Over 20 hours, it is clear that q_{1d} and q_{2d} are both approaching zero, while q_{3d} and q_{4d} are oscillating with a period that is twice the period of our orbit as expected.

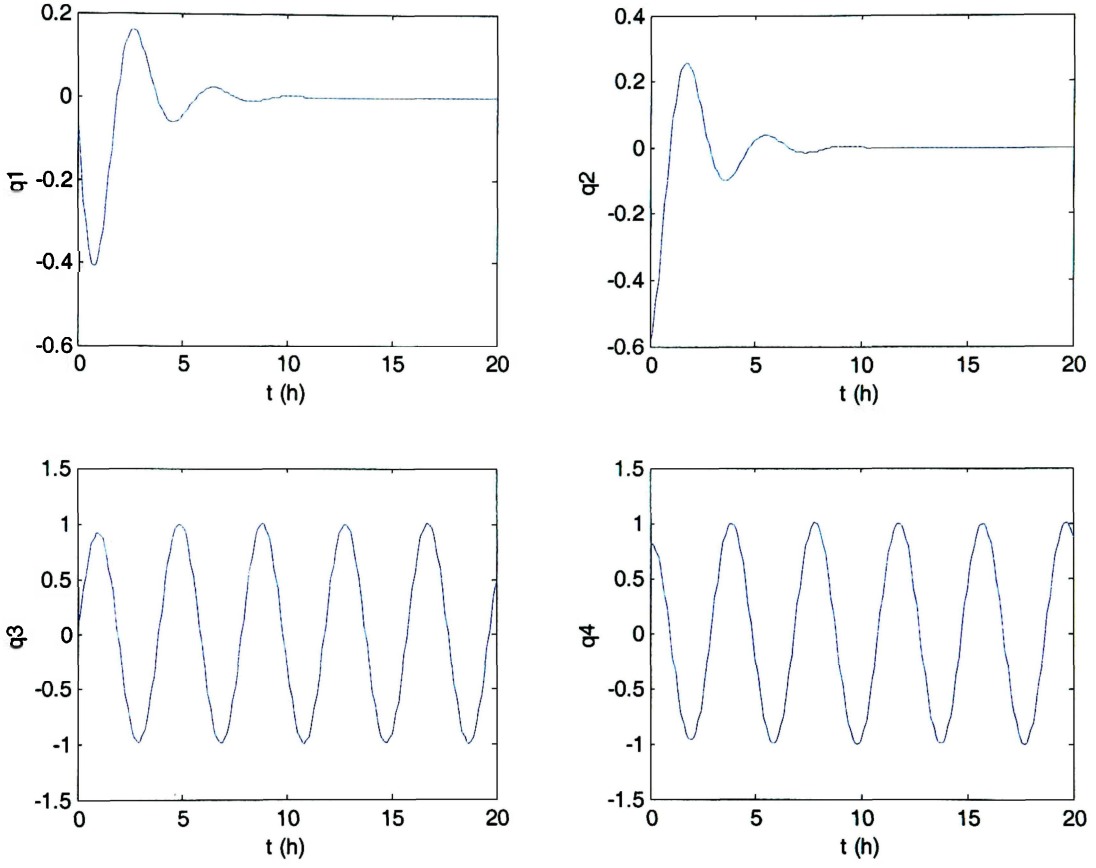


Figure 17 - Quaternions for the Leader Spacecraft

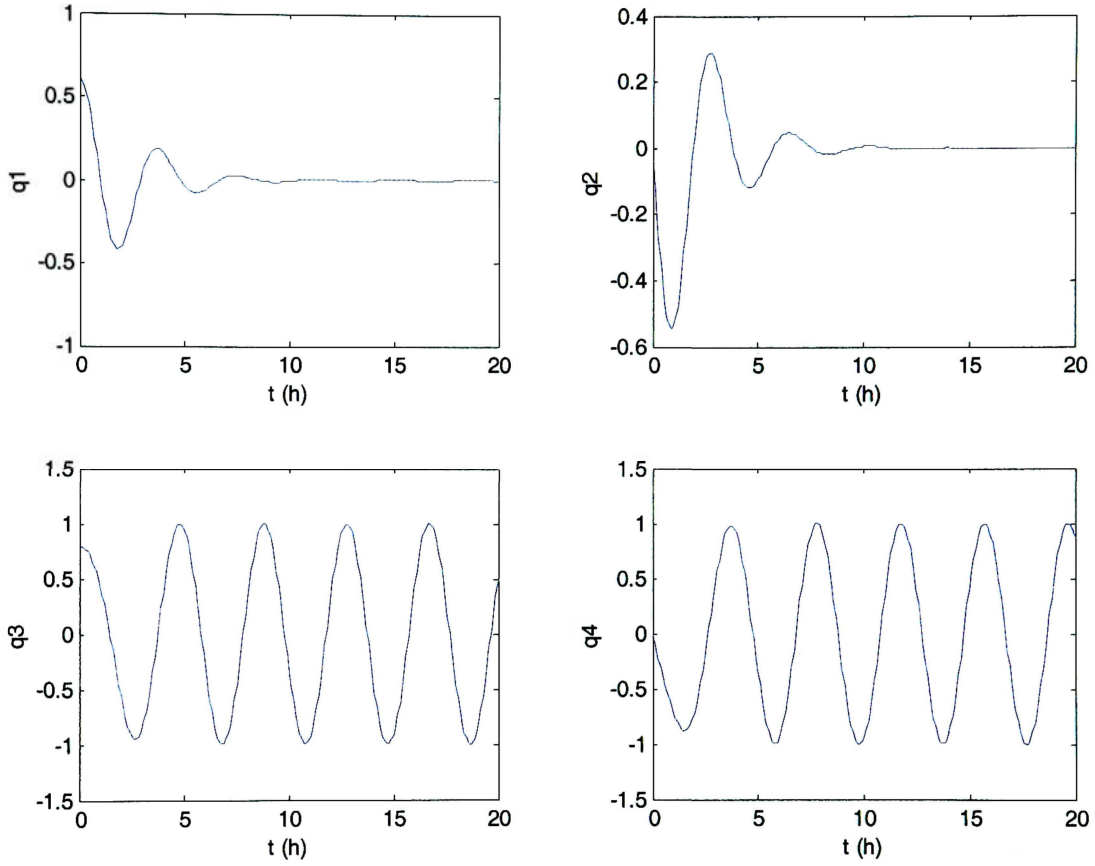


Figure 18 - Quaternions for Follower Spacecraft 1

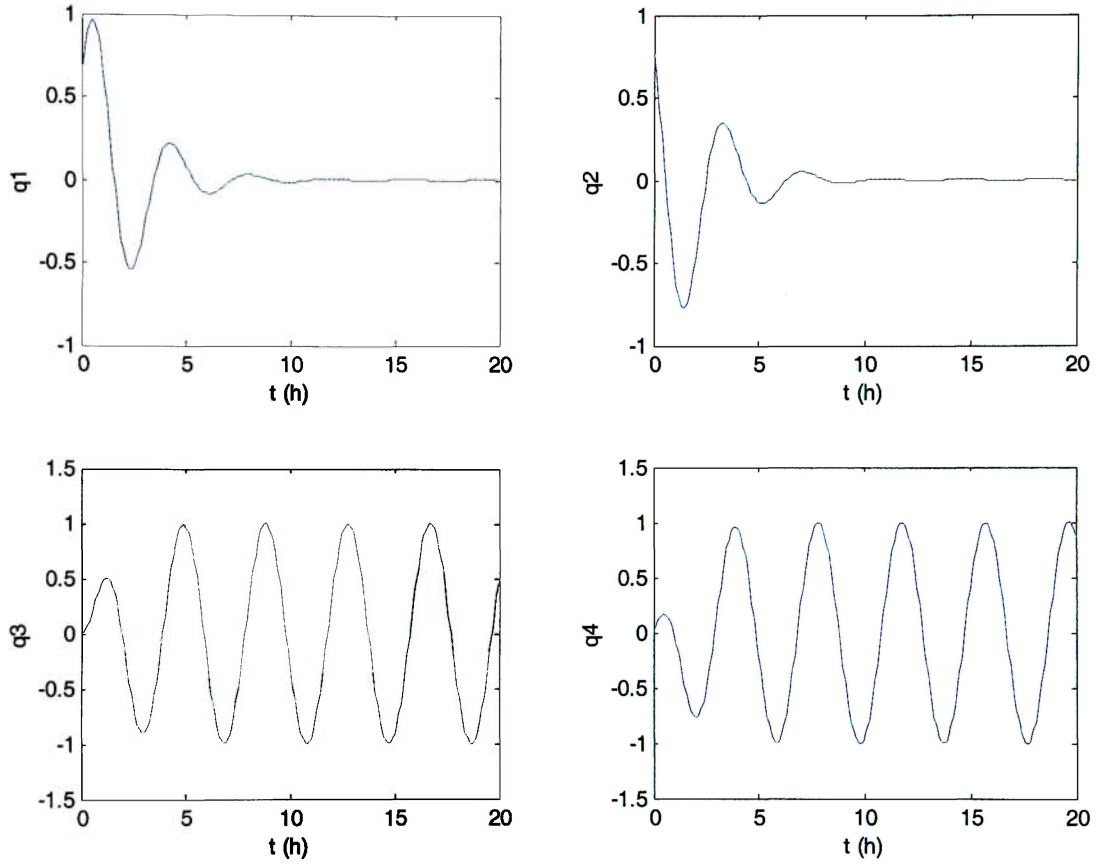


Figure 19 - Quaternions for Follower Spacecraft 2

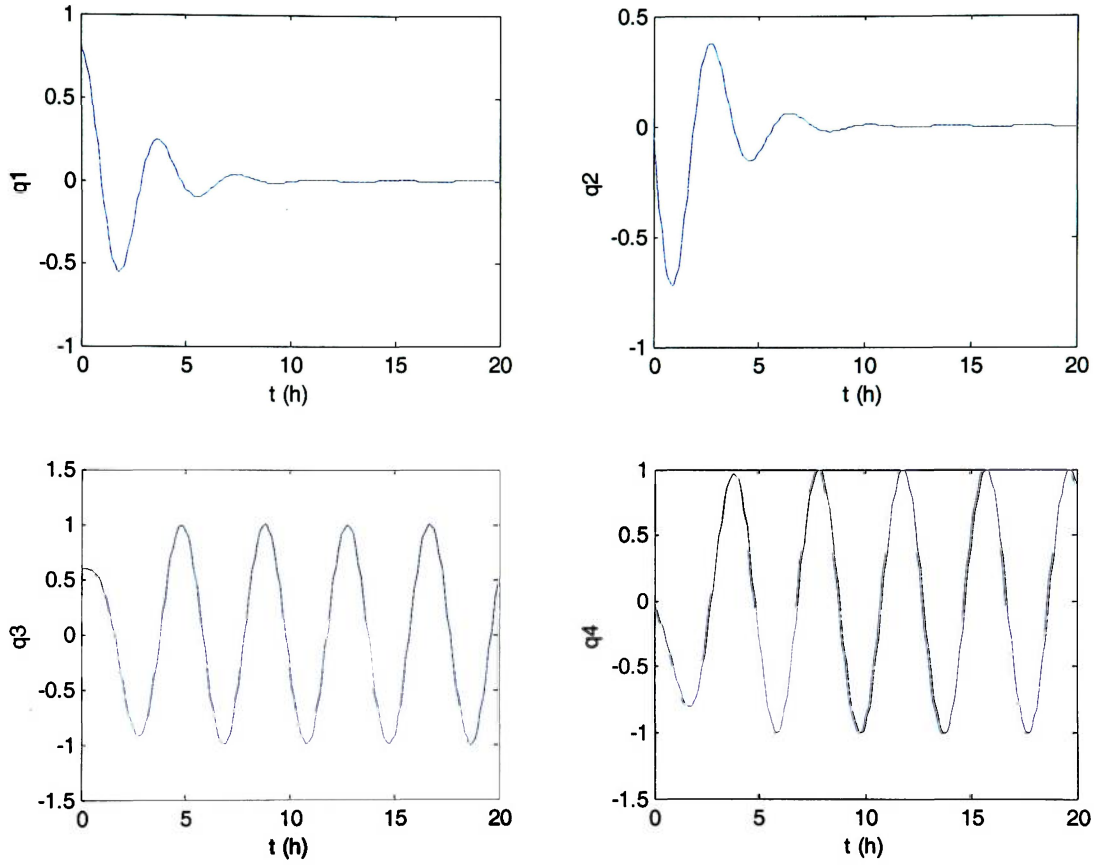


Figure 20 - Quaternions for Follower Spacecraft 3

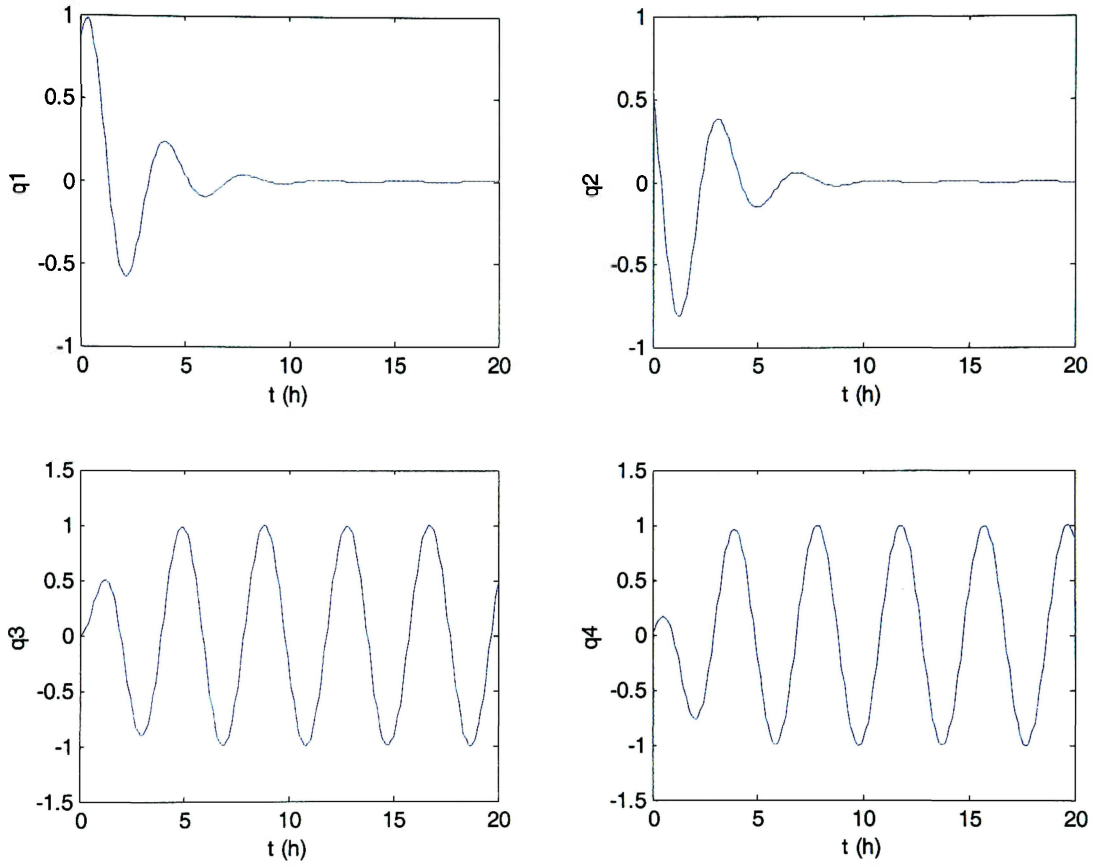


Figure 21 - Quaternions for Follower Spacecraft 4

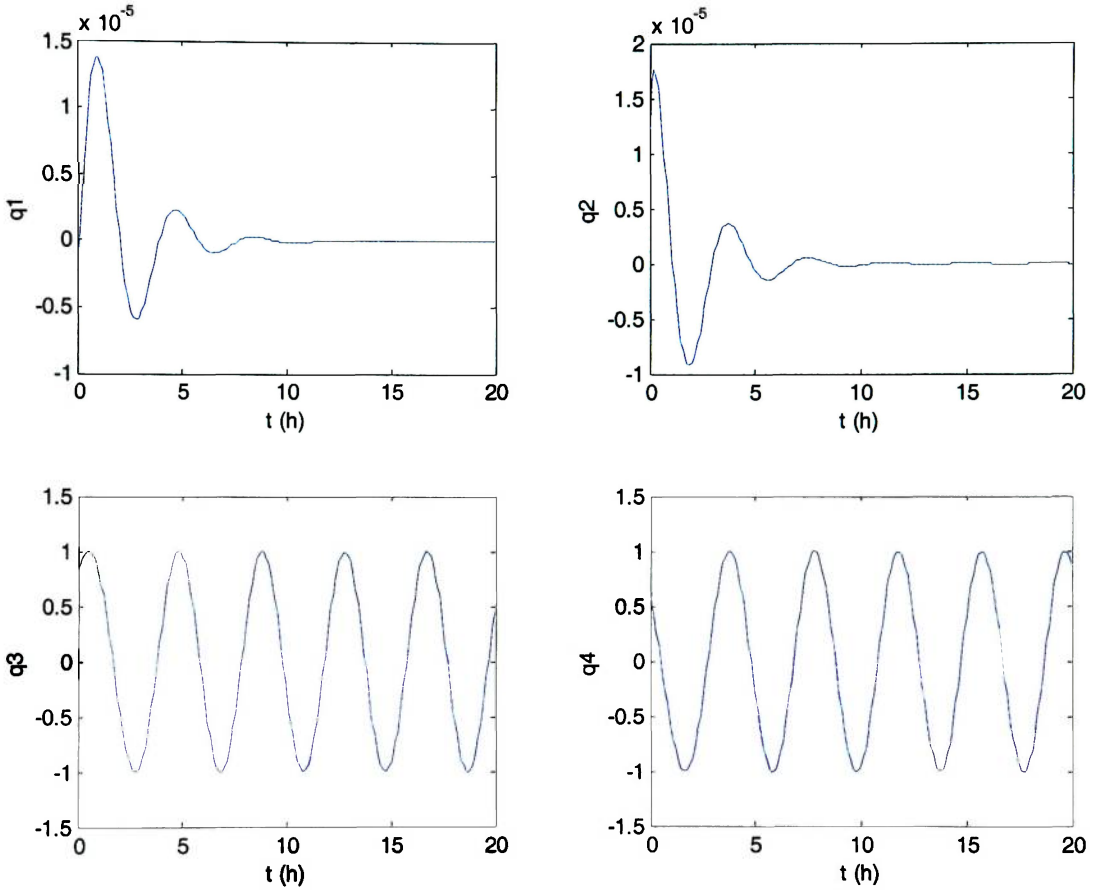


Figure 22 - Quaternions for Follower Spacecraft 5

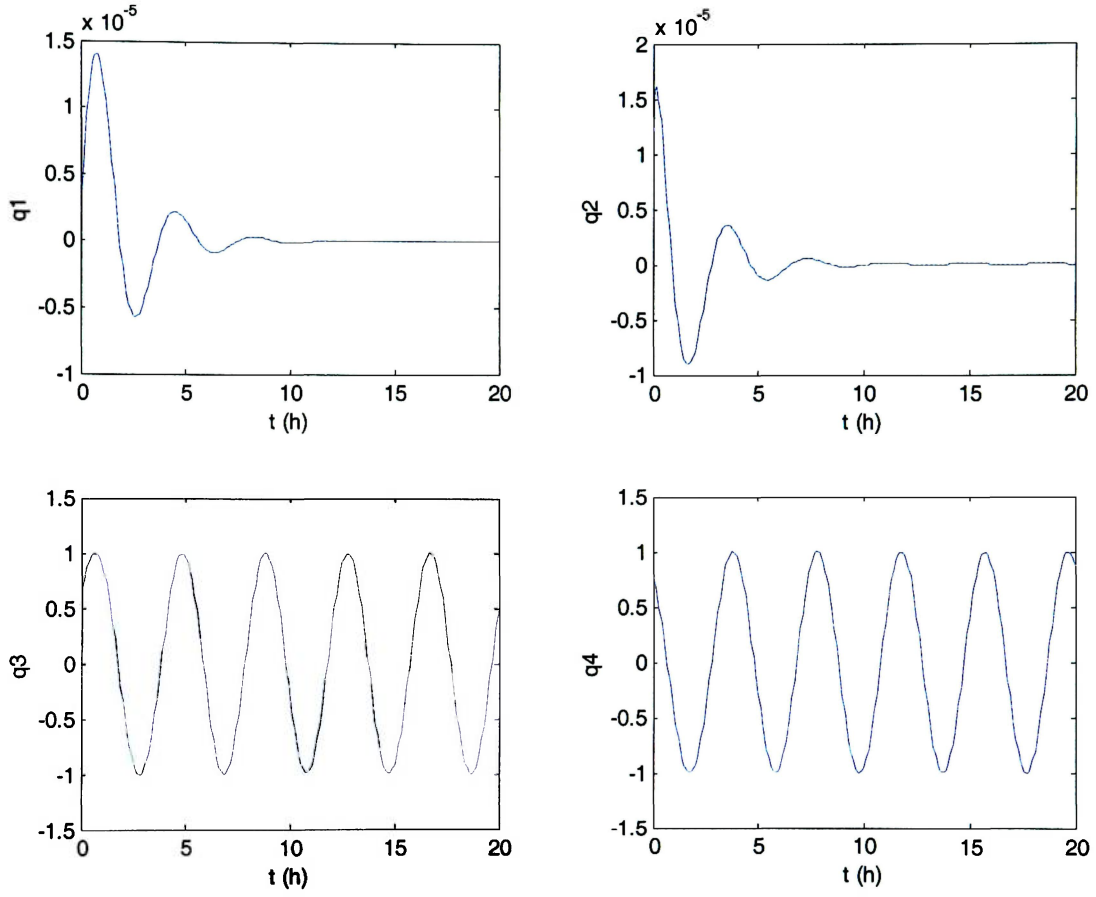


Figure 23 - Quaternions for Follower Spacecraft 6

5.2 Simulation Using STK

To simulate the position and attitude of the formation described earlier, seven spacecraft were created in STK using models from the STK database. Note that these spacecraft merely provide an object for the Matlab data to be applied, any model would suffice. Following the generation of attitude and ephemeris data for all satellites over a specific time period, the data may be loaded into STK. To load an external ephemeris file, the STK External propagator must be selected. This option can be found in the properties window for each spacecraft Basic/Orbit tab. The appropriate files for all seven spacecraft, created by Matlab, were entered in the Basic/Orbit tab. The attitude text files were entered in a similar manner. Under the Basic/Attitude tab, the option to use a pre-computed set of attitude data was selected and each spacecraft attitude text file was loaded into STK.

5.2.1 Frame Transformation

The output data of the translational control program is originally in the frame defined by the leader spacecraft. The data that any simulation will require must be in Earth's reference frame to produce the desired visual in STK. As stated earlier, the basic orbital control of the leader spacecraft was ignored for simplicity. To simulate the leader's orbit, a simple circular orbit was created in the $[X, Y, Z]$ frame as

$$\begin{aligned} X_l &= R\cos(nt) \\ Y_l &= R\sin(nt) \\ Z_l &= 0 \end{aligned} \tag{5.12}$$

where R is the orbital radius, which is set equal to 8000 *km*. The addition of the orbital rate to the time span was required due to the fact that the integration time consists of whole number units. In order for (5.12) to form an acceptable orbit, the whole number time units must incorporate the orbital rate (units of $rads/h$) due to the $\sin(nt)$ and $\cos(nt)$ terms. The localized data for each of the followers was then converted to the $[X,Y,Z]$ frame using the following equations:

$$\begin{aligned} X_i &= X_l + \mathrm{p}_{i1}\sin(nt) + \mathrm{p}_{i2}\cos(nt) \\ Y_i &= Y_l - \mathrm{p}_{i1}\cos(nt) + \mathrm{p}_{i2}\sin(nt) \\ Z_i &= \mathrm{p}_{i3} \end{aligned} \tag{5.13}$$

where $\mathbf{p}_i = [\mathrm{p}_{i1} \quad \mathrm{p}_{i2} \quad \mathrm{p}_{i3}]^T$ is the relative position vector of the ith spacecraft.

5.2.2 Creating File Output for STK

Satellite Tool Kit (STK) allows attitude data and ephemeris data to be input through a simple text file. However, these text files must match a particular format to be recognized by STK. An example attitude text file and ephemeris text file can be found in APPENDIX B and APPENDIX C, respectively. In order to produce these files with Matlab, the Matlab command 'fopen' was used along with 'fprint' and a variety of spacing commands. A loop was used to create a file for each spacecraft and subsequently print the appropriate data to each file (see the attached Matlab code in APPENDIX A).

After transferring all attitude and ephemeris data to STK through text file information, a movie can be generated using STK's Soft VTR application. After opening the properties tab for the 3D window, choosing *Record Animation Every Time Step* enables STK to create an .avi file from the current animation in the 3D window. However, two important

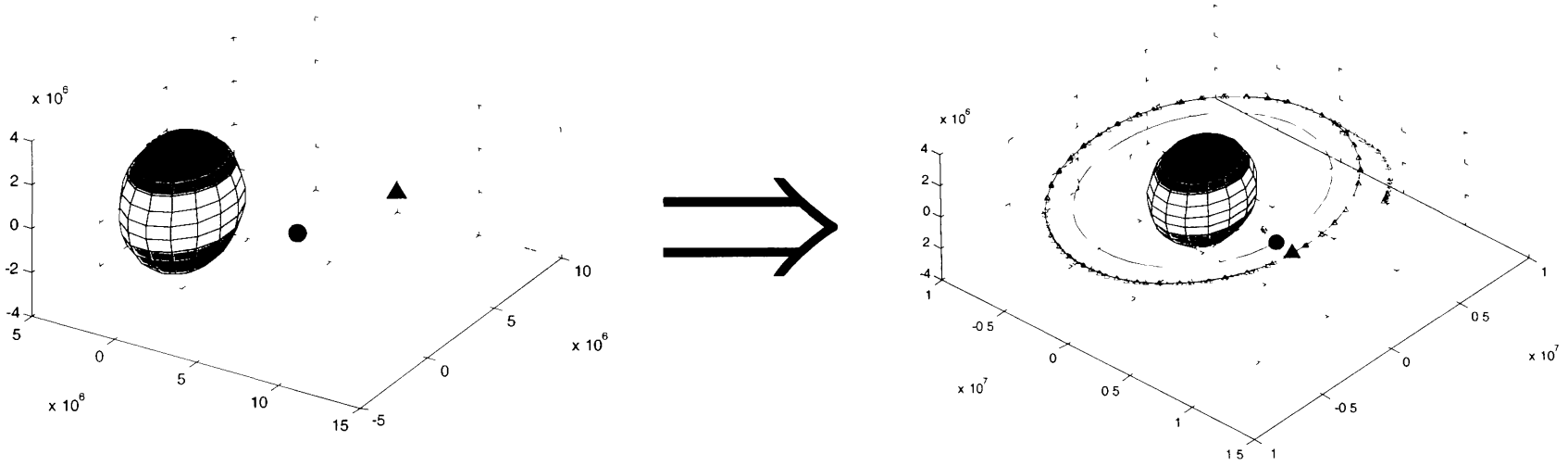
items should be considered when attempting movie creation in STK. First, the STK program allows a choice of which codec to use in order to compress the video file. The codecs used in the formation simulation were DivX and MPEG4. The option of choosing not to compress the file can result in very large file sizes. Second, the 3D window size MUST be set to one of the default sizes found in the 3D window properties tab under Window Properties/Placement (320x240, 640x480 or initial).

5.3 Animations

5.3.1 Matlab Animations

Before simulating the seven spacecraft system, the controlled translational and rotational motions were simulated using only Matlab as in Figure 3, Figure 8 and Figure 9. In addition, an animation was developed in Matlab to visualize translational motion for three spacecraft. Figure 24 shows the initial position in the animation followed by the resulting alignment at the end of the animation.

Figure 24 - Three satellite translational test animation frames



5.3.2 STK Animations

After confirming the effectiveness of the translational and rotational control programs, STK was used to produce 3D animations to help visualize the dynamics of the spacecraft formation. Indeed, once visualized, many issues concerning the motion of the formation were realized. The animations provided a clear view of the motion of the formation with six degrees of freedom, which is much easier to understand than attempting to determine the motion by sifting through Matlab data or attempting to produce a complicated 6 D.O.F animation in Matlab. In fact, the issue of targeting was not realized until after an STK animation was produced. After seeing a visual of the formation with all spacecraft rotating as opposed to focusing together to accomplish a common task, it was obvious some type of targeting needed to be incorporated into the program. The included CD contains all animations produced using STK. Viewing the animation in the STK program allows the user to adjust various parameters such as the time step, angle of view and zoom. However, for presentation purposes, an *.avi file was created so it may be shown on systems without Satellite Tool Kit software. The animations included on the attached CD display the initial creation of the formation described in Chapter IV as well as reconfiguration scenarios.

5.3.3 Formation Versatility

Considering that a major advantage of spacecraft formations is the ability to adjust in failure scenarios and/or to reshape the formation for various experimental purposes, a second animation was created in STK in order to visualize two reconfiguration scenarios.

As before, Matlab was used to generate output files to be imported by STK. Only small changes were needed, such as adding new "desired" position after a certain time period, altering the output of the text file and creating separate integration loops for each different formation. The first rearrangement of the formation is displayed in Figure 25 with the orbit normal out of the page towards the reader. This formation resembles the type that may be used for gradient measurements, such as determining Earth's magnetic field gradients.

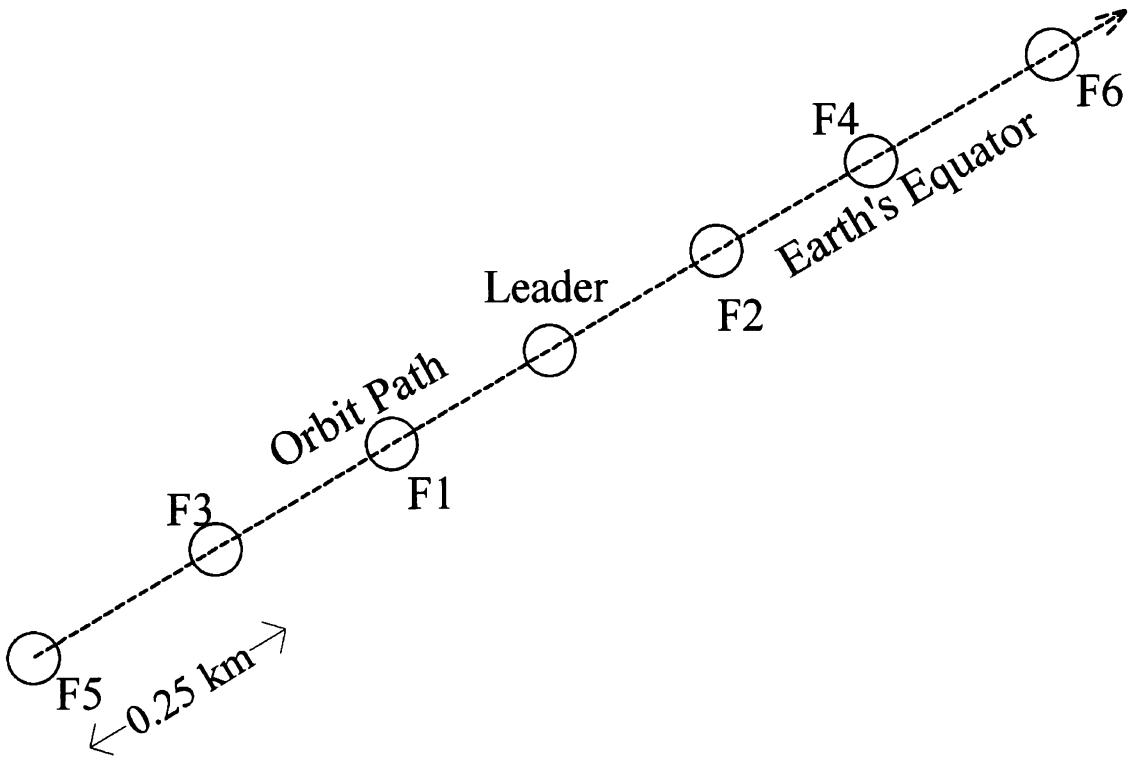


Figure 25 - Altered seven satellite formation pattern

After a short time, a third formation takes shape as shown in Figure 26, again with the orbit normal out of the page towards the reader. This last formation resembles a possible interferometer.

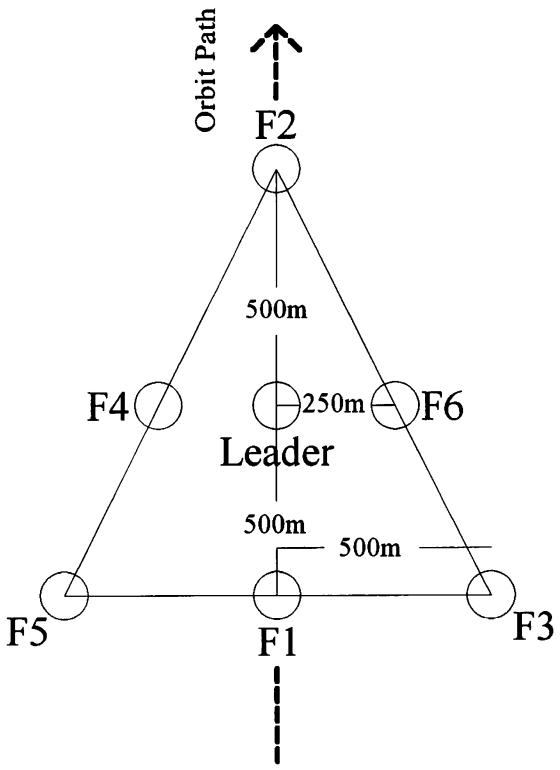


Figure 26 - Final satellite formation

CHAPTER VI

6 CONCLUSION

While some aspects of the thesis proved to be challenging, all of the goals set forth for this thesis have been reached without fail. First, Lyapunov's direct method was used to design control laws for both translational and rotational motion of a formation of spacecraft. Second, these control laws have been tested using a Matlab program to simulate a formation over time. Third, these programs have been combined to create a simulation of both translational and rotational control of the formations. Finally, Satellite Tool Kit has been used to create a 3-dimensional view of this simulated control in order to easily visualize its effectiveness.

A variety of issues regarding formation control have not been addressed in this thesis. One major factor ignored in this thesis involves the actual physical controls that would be required to perform position and attitude adjustments. The actual mechanical control system would apply certain restraints on the system and would likely result in varying time periods until the desired positions are reached, depending on mission constraints. However, due to the robust nature of Lyapunov's method of control, any additional terms, linear or non-linear, could be dealt with.

Another issue involves fuel economy. In order for the follower spacecraft to maintain their relative positions, a large amount of energy expenditure would be needed due to their changing or "artificial" orbits. Considering the formation described in this thesis, the only unadjusted orbit would be that of the leader. The follower spacecraft would need

to continuously readjust to maintain their relative positions. This constant readjustment could be quite costly in terms of fuel economy.

One possible solution would be to restrict the size of the formation so that the entire formation would follow the same basic orbit, give or take a few meters. Another solution could be to restrict each spacecraft in the formation to attempt to match a natural orbit. Both of these solutions might be further optimized by the use of micro/nanosatellites, where a smaller amount of fuel is necessary for maneuvers.

While this thesis tackles multiple aspects of spacecraft formation flight and simulation, there are many avenues of future study. As mentioned previously, the application of these methods to physical control systems may be explored. Also, there is the possibility of collision during formation creation. The program created in this thesis does not incorporate that possibility, and onboard sensors are assumed to help prevent such collisions among spacecraft in the group. Finally, a large area of study is the identification and design of various formations to perform specific functions.

REFERENCES

[1] C.A. Bailey, T.W. McLain and, R.W. Beard, "Fuel-Saving Strategies for Dual Spacecraft Interferometry Missions," Journal of Astronautical Sciences, Vol.49, No.3, 2001, pp.469-488.

[2] R.W. Beard, J. Lawton, and F.Y. Hadaegh, "A Coordination Architecture for Spacecraft Formation Control," IEEE Transactions on Control Systems Technology, Vol.9, No.6, 2001, pp.777-790.

[3] J.D. Boskovic, S.-M. Li, and R.K. Mehra, "Globally Stable Adaptive Tracking Control Design for Spacecraft under Input Saturation," Proceedings of IEEE Conference on Decision and Control, 1999, pp.1952-1957.

[4] A. Decou, "Multiple Spacecraft Optical Interferometry Preliminary Feasibility Assessment," Internal Report D-8811, Jet Propulsion Laboratory, Pasedena, CA, August 1991.

[5] P. Gurfil and N.J. Kasdin, "Nonlinear Low-Thrust Lyapunov-Based Control of Spacecraft Formations," Proceedings of the American Control Conference, 2003, pp. 1758-1763.

[6] P.C. Hughes, Spacecraft Attitude Dynamics, John Wiley & Sons, Inc., New York, 1986.

[7] I.I. Hussein, D.J. Scheeres, D.C. Hyland, "Control of a Satellite Formation For Imaging Applications," Proceedings of the American Control Conference, 2003, pp.308-313.

[8] J. Lawton and R.W. Beard, "Model Independent Eigenaxis Maneuvers Using Quaternion Feedback," Proceedings of the American Control Conference, 2001, pp.2339-2344.

[9] J. Lawton, R.W. Beard and F.Y. Hadaegh, "Elementary Attitude Formation Maneuvers via Leader-Following and Behavior-Based Control," Proceedings of AIAA Guidance, Navigation and Control Conference, 2000, pp.1-11.

[10] D. Lee and P.Y. Li, "Formation and Maneuver Control of Multiple Spacecraft," Proceedings of the American Control Conference, 2003, pp.278-283.

[11] M.S. de Queiroz, Q. Yan, G. Yang, and V. Kapila, "Global Output Feedback Tracking Control of Spacecraft Formation Flying with Parametric Uncertainty," Proceedings of IEEE Conference on Decision and Control, 1999, pp. 584-589.

[12] W. Ren and R.W. Beard, "A Decentralized Scheme for Spacecraft Formation Flying via the Virtual Structure Approach," Proceedings of the American Control Conference, 2003, pp.1746-1751.

[13] D.P. Scharf, F.Y. Hadaegh, and S.R. Ploen, "A Survey of Spacecraft Formation Flying Guidance and Control (Part I): Guidance," Proceedings of the American Control Conference, 2003, pp.1733-1739.

[14] A. Serrani, "Robust Coordinated Control of Satellite Formations Subject to Gravity Perturbations," Proceedings of the American Control Conference, 2003, pp.302-307.

[15] A. Sparks, "Satellite Formation Keeping Control in the Presence of Gravity Perturbations," Proceedings of the American Control Conference, 2000, pp.844-848.

[16] R.S. Smith and F.Y. Hadaegh, "Control Topologies for Deep Space Formation Flying Spacecraft," Proceedings of the American Control Conference, 2002, pp.2836-2841.

[17] M. Tillerson, L. Breger, and J.P. How, "Distributed Coordination and Control of Formation Flying Spacecraft," Proceedings of the American Control Conference, 2003, pp.1740-1745.

[18] P.K.C. Wang and F.Y. Hadaegh, "Coordination and Control of Multiple Micro-Spacecraft Moving in Formation," Journal of Astronautical Sciences, Vol. 44, No. 3, 1996, pp.315-355.

[19] P.K.C. Wang, F.Y. Hadaegh and L. Kau,"Synchronized Formation Rotation and Attitude Control of Multiple Free-Flying Spacecraft," AIAA Journal on Guidance, Control and Dynamics, Vol. 22, No. 1, 1999, pp.28-35.

[20] B. Wie, Space Vehicle Dynamics and Control, American Institute of Aeronautics and Astronautics, Inc., Wright-Patterson AFB, Ohio, 1998.

[21] K. Yamanaka, "Simultaneous Translation and Rotation Control Law for Formation Flying Satellites," Proceedings of AIAA Guidance, Navigation and Control Conference, 2000, pp.1-10.

[22] Q. Yan, G. Yang, V. Kapila and M.S. Queiroz, "Nonlinear Dynamics and Output Feedback Control of Multiple Spacecraft in Elliptical Orbits," Proceedings of the American Control Conference, 2000, pp.839-843.

[23] H.-H. Yeh and A. Sparks, "Geometry and Control of Satellite Formations," Proceedings of the American Control Conference, 2000, pp.384-388.

APPENDIX A

MATLAB SIMULATION CODE

```
% This program simulates rotational control of microsatellites in formation
% adapted from a paper from Koji Yamanaka 'Simultaneous Translation and
% Rotation Control Law for Formation Flying Satellites' and a paper from Koji
%Yamanaka 'Nonlinear Dynamics and Output Feedback Control of Multiple Spacecraft
%in Elliptical Orbits'
clear
global k c J1 J2 J3 p B K el pos nn rl R Rq thd thdd mu mf r
%-------------------------------------------------------------
% GENERAL VARIABLES
%-------------------------------------------------------------
% Number of satellites in formation
n=7;
s=n;
% Leader orbit radius
r=8000;

% Orbit parameters
mu=398600*3600*3600;

% Leader orbit period, T=Orbital period
T=(2*pi/sqrt(mu))*r^(3/2);

% Orbital rate
nn=2*pi/T;

%-------------------------------------------------------------
% ROTATIONAL CONTROL VARIABLES
%-------------------------------------------------------------
k=10;
c=10;

% The initial inertia tensor for all s/c
J1=33;
J2=33;
J3=50;

% Initial angular velocities
w1=1/3600;
w2=1/3600;
w3=1/3600;

w=[w1 w2 w3]';
```

```
% Initial leader quaternions
q10=0;
q20=-.6;
q30=0;
q40=.8;

% Follower 1 quaternions
q1f1=0.6;
q2f1=0;
q3f1=0.8;
q4f1=0;

% Follower 2 quaternions
q1f2=0.6;
q2f2=0.8;
q3f2=0;
q4f2=0;

% Follower 3 quaternions
q1f3=0.8;
q2f3=0;
q3f3=0.6;
q4f3=0;

% Follower 4 quaternions
q1f4=0.8;
q2f4=0.6;
q3f4=0;
q4f4=0;

% Follower 5 quaternions
q1f5=0;
q2f5=0;
q3f5=0.8;
q4f5=0.6;

% Follower 6 quaternions
q1f6=0;
q2f6=0;
q3f6=0.6;
q4f6=0.8;

% At time zero vector
x0rot=[q10 q20 q30 q40 w1 w2 w3;
     q1f1 q2f1 q3f1 q4f1 w1 w2 w3;
     q1f2 q2f2 q3f2 q4f2 w1 w2 w3;
```

```
    q1f3 q2f3 q3f3 q4f3 w1 w2 w3;
    q1f4 q2f4 q3f4 q4f4 w1 w2 w3;
    q1f5 q2f5 q3f5 q4f5 w1 w2 w3;
    q1f6 q2f6 q3f6 q4f6 w1 w2 w3]';
Xf=0;
%-------------------------------------------------------------
% TRANSLATIONAL CONTROL VARIABLES
%-------------------------------------------------------------

%Control gains
B=[2 0 0;
  0 2 0;
  0 0 2];
K=[1 0 0;
  0 1 0;
  0 0 1];

%Constants
mf=100;
el=0; %Leader eccentricity

%Desired Positions
pos1=[500e-3 0 0]';
pos2=[-500e-3 0 0]';
pos3=[250e-3 0 -433e-3]';
pos4=[-250e-3 0 433e-3]';
pos5=[250e-3 0 433e-3]';
pos6=[-250e-3 0 -433e-3]';
dpos=horzcat(pos1,pos2,pos3,pos4,pos5,pos6);

% % Desired Positions II
pos1er=[250e-3 0 0]';
pos2er=[-250e-3 0 0]';
pos3er=[500e-3 0 0]';
pos4er=[-500e-3 0 0]';
pos5er=[750e-3 0 0]';
pos6er=[-750e-3 0 0]';
dposer=horzcat(pos1er,pos2er,pos3er,pos4er,pos5er,pos6er);
%
% % Desired Positions II
pos1er2=[500 0 0]';
pos2er2=[-500 0 0]';
pos3er2=[500 0 -500]';
pos4er2=[0 0 250]';
pos5er2=[500 0 500]';
pos6er2=[0 0 -250]';
```

```
dposer2=horzcat(pos1er2,pos2er2,pos3er2,pos4er2,pos5er2,pos6er2);

%Initial conditions
%-----------
%Satellite 1
%-----------
th1=0; %radians
xi1=-.5; %kmeters
yi1=.8;
zi1=-.1;
xdi1=-.136; %km/h
ydi1=0;
zdi1=0.04;
x0t1=[xi1 yi1 zi1 xdi1 ydi1 zdi1 th1];
%-----------
%Satellite 2
%-----------
th2=0;
xi2=.7; %kmeters
yi2=.5;
zi2=.2;
xdi2=-.136; %km/h
ydi2=0;
zdi2=0.04;
x0t2=[xi2 yi2 zi2 xdi2 ydi2 zdi2 th2];
%-----------
%Satellite 3
%-----------
th3=0;
xi3=.2; %km
yi3=1;
zi3=1;
xdi3=-.136; %km/h
ydi3=0;
zdi3=0.04;
x0t3=[xi3 yi3 zi3 xdi3 ydi3 zdi3 th3];
%-----------
%Satellite 4
%-----------
th4=0;
xi4=.6; %km
yi4=1;
zi4=-1;
xdi4=-.136; %km/h
ydi4=0;
zdi4=0.04;
```

```
x0t4=[xi4 yi4 zi4 xdi4 ydi4 zdi4 th4];
%----------
%Satellite 5
%----------
th5=0;
xi5=.4; %km
yi5=.8;
zi5=1;
xdi5=-.136; %meters/sec
ydi5=0;
zdi5=0.04;
x0t5=[xi5 yi5 zi5 xdi5 ydi5 zdi5 th5];
%----------
%Satellite 6
%----------
th6=0;
xi6=.5; %meters
yi6=.2;
zi6=.25;
xdi6=-.136; %km/h
ydi6=0;
zdi6=0.04;
x0t6=[xi6 yi6 zi6 xdi6 ydi6 zdi6 th6];
%----------
%Combine
%----------
x0t=vertcat(x0t1,x0t2,x0t3,x0t4,x0t5,x0t6)';
tt=0:.1:20;
%-------------------------------------------------------------
% ROTATIONAL INTEGRATION
%-------------------------------------------------------------
for j=1:n
   xir=x0rot(:,j);
   % Integrate
   for t=1:201
   [Xir]=RK4('STKattFunc2',.1*(t-1),xir,.1);
   Xr(:,t,j)=Xir;
   xir=Xir;
end
end
%_____________________________________________________________________
% Create attitude quaternion files for transfer to Satellite Tool Kit (STK)
%_____________________________________________________________________
for j=1:n
   if j==1
   att=fopen('Leader.a','w');
```

```
    end
    if j==2
    att=fopen('Flwr1.a','w');
    end
    if j==3
    att=fopen('Flwr2.a','w');
    end
    if j==4
    att=fopen('Flwr3.a','w');
    end
    if j==5
    att=fopen('Flwr4.a','w');
    end
    if j==6
    att=fopen('Flwr5.a','w');
    end
    if j==7
    att=fopen('Flwr6.a','w');
    end
fprintf(att,'stk.v.5.0\n\n');
fprintf(att,'BEGIN Attitude\n\n');
fprintf(att,'ScenarioEpoch\t\t 1 Jun 2005 12:00:00.00\n');
fprintf(att,'NumberOfAttitudePoints\t');
fprintf(att,'%d',t);
fprintf(att,'\n');
fprintf(att,'BlockingFactor\t\t');
fprintf(att,'%d',20);
fprintf(att,'\n');
fprintf(att,'InterpolationOrder\t');
fprintf(att,'%d',1);
fprintf(att,'\n');
fprintf(att,'CentralBody\t\t\t Earth\n');
fprintf(att,'CoordinateAxes\t\t J2000\n\n');
fprintf(att,'AttitudeTimeQuaternions\n\n');
for h=1:t
      fprintf(att,'\t%-.4f \t%-.3f \t%-.3f \t%-.3f \t%-.3f
\n',h,Xr(1,h,j),Xr(2,h,j),Xr(3,h,j),Xr(4,h,j));
end
fprintf(att,'\n');
fprintf(att,'END Attitude');
status = fclose(att);
end
%-------------------------------------------------------------
% TRANSLATIONAL INTEGRATION
%-------------------------------------------------------------
for m=1:n-1
```

```
  %Select initial satellite conditions from total condition matrix
  xit=x0t(:,m);
  %Select desired position from desired position matrix
  pos=dpos(:,m);
  % Integrate
  for t=1:201
  [Xit]=RK4('STKtransFunc',0.1*(t-1),xit,0.1);
  xit=Xit;
  rs(:,t,m)=Xit;
  end
  tspan=1:t;
  %Create arbitrary circular orbit for the leader using the orbital rate
  Rx=r.*cos((nn)*tspan);
  Ry=r.*sin((nn)*tspan);
  Rz=0.*tspan;
  %Convert local leader frame to earth XYZ frame for follower
  Xit=Xit';
  rx(1,:,m)=Rx+sin(nn*tspan).*rs(1,:,m)+cos(nn*tspan).*rs(2,:,m);
  ry(1,:,m)=Ry-cos(nn*tspan).*rs(1,:,m)+sin(nn*tspan).*rs(2,:,m);
  rz(1,:,m)=rs(3,:,m);
  rsats(:,:,m)=vertcat(rx(1,:,m),ry(1,:,m),rz(1,:,m)); %Full position matrix in Earth frame

end

for m=1:n-1
  %Select initial satellite conditions from total condition matrix
  xit=rs(:,20,m);
  %Select desired position from desired position matrix
  pos=dposer(:,m);
  % Integrate
  for t=21:35
  [Xit]=RK4('STKtransFunc',t,xit,1);
  xit=Xit;
  rs(:,t,m)=Xit;
  end
  tspan=1:t;
  %Create arbitrary circular orbit for the leader using the orbital rate
  Rx=r.*cos((nn)*tspan);
  Ry=r.*sin((nn)*tspan);
  Rz=0.*tspan;
  %Convert local leader frame to earth XYZ frame for follower
  Xit=Xit';
  rxer(1,:,m)=Rx+sin(nn*tspan).*rs(1,:,m)+cos(nn*tspan).*rs(2,:,m);
  ryer(1,:,m)=Ry-cos(nn*tspan).*rs(1,:,m)+sin(nn*tspan).*rs(2,:,m);
  rzer(1,:,m)=rs(3,:,m);
```

```
   rsatser(:,:,m)=vertcat(rxer(1,:,m),ryer(1,:,m),rzer(1,:,m)); %Full position matrix in
Earth frame

end

for m=1:n-1
   %Select initial satellite conditions from total condition matrix
   xit=rs(:,35,m);
   %Select desired position from desired position matrix
   pos=dposer2(:,m);
   % Integrate
   for t=36:200
   [Xit]=RK4('STKtransFunc',t,xit,1);
   xit=Xit;
   rs(:,t,m)=Xit;
   end
   tspan=1:t;
   %Create arbitrary circular orbit for the leader using the orbital rate
   Rx=r.*cos((nn)*tspan);
   Ry=r.*sin((nn)*tspan);
   Rz=0.*tspan;
   %Convert local leader frame to earth XYZ frame for follower
   Xit=Xit';
   rxer2(1,:,m)=Rx+sin(nn*tspan).*rs(1,:,m)+cos(nn*tspan).*rs(2,:,m);
   ryer2(1,:,m)=Ry-cos(nn*tspan).*rs(1,:,m)+sin(nn*tspan).*rs(2,:,m);
   rzer2(1,:,m)=rs(3,:,m);
   rsatser2(:,:,m)=vertcat(rxer2(1,:,m),ryer2(1,:,m),rzer2(1,:,m)); %Full position matrix in
Earth frame

end
%_____________________________________________________________________
% % Create ephemeris(time/position) files for transfer to Satellite Tool Kit (STK)
%_____________________________________________________________________
for j=0:6
   if j==0
      eph=fopen('Leader.e','w');
   end
   if j==1
      eph=fopen('Flwr1.e','w');
   end
   if j==2
      eph=fopen('Flwr2.e','w');
   end
   if j==3
      eph=fopen('Flwr3.e','w');
   end
```

```
   if j==4
      eph=fopen('Flwr4.e','w');
   end
   if j==5
      eph=fopen('Flwr5.e','w');
   end
   if j==6
      eph=fopen('Flwr6.e','w');
   end
fprintf(eph,'stk.v.5.0\n\n');
fprintf(eph,'BEGIN Ephemeris\n\n');
fprintf(eph,'ScenarioEpoch\t\t 1 Jun 2005 12:00:00.00\n');
fprintf(eph,'CoordinateSystem\t J2000');
fprintf(eph,'\n');
fprintf(eph,'DistanceUnit\t Meters');
fprintf(eph,'\n');
fprintf(eph,'NumberOfEphemerisPoints\t');
fprintf(eph,'%d',t);
fprintf(eph,'\n');
fprintf(eph,'CentralBody\t\t\t Earth\n');
fprintf(eph,'EphemerisTimePos\n\n');
for h=1:20
   if j>0
      fprintf(eph,'\t%-.4f \t%-.3f \t%-.3f \t%-.1f \n',h,rsats(1,h,j),rsats(2,h,j),rsats(3,h,j));
   end
end
for h=21:35
   if j>0
      fprintf(eph,'\t%-.4f \t%-.3f \t%-.3f \t%-.1f
\n',h,rsatser(1,h,j),rsatser(2,h,j),rsatser(3,h,j));
   end
end
for h=36:t
   if j>0
      fprintf(eph,'\t%-.4f \t%-.3f \t%-.3f \t%-.1f
\n',h,rsatser2(1,h,j),rsatser2(2,h,j),rsatser2(3,h,j));
   end
end

for h=1:t
   if j==0
      fprintf(eph,'\t%-.4f \t%-.3f \t%-.3f \t%-.1f \n',h,Rx(1,h),Ry(1,h),Rz(1,h));
   end
end

fprintf(eph,'\n');
```

```
fprintf(eph,'END Ephemeris');
status = fclose(eph);
end
i=0;
j=0;
%-------------------------------------------------------------
% ROTATION PLOTS
%-------------------------------------------------------------
j=0
hold on
for j=1:7
figure(j)
subplot(2,2,1)
plot(tt,Xr(1,:,j))
xlabel('t (h)')
ylabel('q1')
subplot(2,2,2)
plot(tt,Xr(2,:,j))
xlabel('t (h)')
ylabel('q2')
subplot(2,2,3)
plot(tt,Xr(3,:,j))
xlabel('t (h)')
ylabel('q3')
subplot(2,2,4)
plot(tt,Xr(4,:,j))
xlabel('t (h)')
ylabel('q4')
end
% %-------------------------------------------------------------
% TRANSLATION PLOTS
%-------------------------------------------------------------
for j=8:13
figure(j)
subplot(3,1,1)
plot(tt,rs(1,:,j-7))
ylabel('x (km)')
subplot(3,1,2)
plot(tt,rs(2,:,j-7))
ylabel('y (km)')
subplot(3,1,3)
plot(tt,rs(3,:,j-7))
xlabel('time (h)')
ylabel('z (km)')
end
```

TRANSLATION CONTROL FUNCTION CODE

```
function xd=f(t,x)

global B K el mf mu s p pos rl r nn

%Variable set-up
rl=r*(1-el^2)/(1+el*cos(x(1)));  %m
R=sqrt(rl^2);
Rq=sqrt(x(1)^2+(x(2)+rl)^2+x(3)^2);
thd=(nn*(1+el*cos(x(1)))^2)/(1-el^2)^(3/2);
thdd=(-2*nn^2*el*(1+el*cos(x(1)))^3*sin(x(1)))/(1-el^2)^3;
N1=mu*x(1)/Rq^3-(thd^2*x(1)+thdd*x(2));
N2=mu*(((x(2)+R)/Rq^3)-1/R^2)-(thd^2*x(2)-thdd*x(1));
N3=mu*x(3)/Rq^3;
N=[N1 N2 N3]';
C=2*thd*[0 -1 0;1 0 0;0 0 0];
q=[x(1) x(2) x(3)]';
qd=[x(4) x(5) x(6)]';

%Control
uf=mf*((C*qd+N)-B*qd-K*(q-pos));

% x-dot values
xd=[x(4);
   x(5);
   x(6);
   uf(1)/mf+2*thd*x(5)-N1;
   uf(2)/mf-2*thd*x(4)-N2;
   uf(3)/mf-N3;
   thd];
```

ROTATION CONTROL FUNCTION CODE

```
function xd=f(t,x)
global k c J1 J2 J3 nn

% Orbital rate used from main file

% Adjust desired quaternions to continuously match the orbital period
qd=[0 0 sin(nn*t/2) cos(nn*t/2)]';

% Attitude error quaternion relationship
Qd=[qd(4) qd(3) -qd(2) -qd(1);
    -qd(3) qd(4) qd(1) -qd(2);
    qd(2) -qd(1) qd(4) -qd(3);
    qd(1) qd(2) qd(3) qd(4)];
J=[J1 0 0;0 J2 0;0 0 J3];
q(1)=x(1);
q(2)=x(2);
q(3)=x(3);
q(4)=x(4);
w(1)=x(5);
w(2)=x(6);
w(3)=x(7);

q=[q(1) q(2) q(3) q(4)]';

Sw=[0 -w(3) w(2);w(3) 0 -w(1);-w(2) w(1) 0];
qe=Qd*q;
qev=[qe(1) qe(2) qe(3)]';
w=[w(1) w(2) w(3)]';
wd=[0 0 nn]';
% Control
u=-k*J*qev-c*J*(w-wd)+Sw*J*w;

% x-dot values
xd=0.5*[x(7)*x(2)-x(6)*x(3)+x(5)*x(4);
    x(4)*x(6)-x(7)*x(1)+x(3)*x(5);
    x(4)*x(7)+x(6)*x(1)-x(5)*x(2);
    -x(1)*x(5)-x(6)*x(2)-x(7)*x(3);
    2*u(1)/J1;
    2*u(2)/J2;
    2*u(3)/J3];
```

APPENDIX B

SAMPLE ATTITUDE FILE USED BY SATTELITE TOOL KIT

```
stk.v.5.0

BEGIN Attitude

ScenarioEpoch            1 Jun 2005 12:00:15.00
NumberOfAttitudePoints   75
BlockingFactor           20
InterpolationOrder       1
CentralBody              Earth
CoordinateAxes           J2000

AttitudeTimeQuaternions

      1.0000      0.725       0.124       0.500       0.458
      2.0000      0.612       0.268       0.689       0.284
      3.0000      0.465       0.419       0.718       0.310
      4.0000      0.311       0.548       0.635       0.450
      5.0000      0.160       0.618       0.465       0.616
      6.0000      0.033       0.618       0.258       0.744
      7.0000      -0.052      0.572       0.073       0.818
      8.0000      -0.092      0.516       -0.056      0.852
      9.0000      -0.098      0.474       -0.123      0.868
      10.0000     -0.083      0.454       -0.140      0.878
      11.0000     -0.058      0.452       -0.123      0.884
      12.0000     -0.031      0.460       -0.090      0.885
      13.0000     -0.009      0.473       -0.052      0.881
      14.0000     0.006       0.486       -0.019      0.876
      15.0000     0.014       0.496       0.005       0.870
      16.0000     0.016       0.503       0.018       0.866
      17.0000     0.014       0.507       0.023       0.864
      18.0000     0.011       0.508       0.021       0.863
      19.0000     0.006       0.507       0.016       0.864
      20.0000     0.002       0.506       0.010       0.865
      21.0000     -0.000      0.504       0.004       0.866
      22.0000     -0.002      0.502       0.000       0.867
      23.0000     -0.003      0.501       -0.003      0.868
      24.0000     -0.002      0.500       -0.004      0.868
      25.0000     -0.002      0.500       -0.004      0.868
      26.0000     -0.001      0.500       -0.003      0.868
      27.0000     -0.001      0.500       -0.002      0.868
      28.0000     -0.000      0.500       -0.001      0.868
      29.0000     0.000       0.501       -0.000      0.868
      30.0000     0.000       0.501       0.000       0.867
      31.0000     0.000       0.501       0.001       0.867
      32.0000     0.000       0.501       0.001       0.867
      33.0000     0.000       0.501       0.001       0.867
      34.0000     0.000       0.501       0.000       0.867
      35.0000     0.000       0.501       0.000       0.867

END Attitude
```

APPENDIX C

SAMPLE EPHEMERIS FILE USED BY SATTELITE TOOL KIT

```
stk.v.5.0

BEGIN Ephemeris

ScenarioEpoch            1 Jun 2005 12:00:00.00
CoordinateSystem   J2000
DistanceUnit       Meters
NumberOfEphemerisPoints 200
CentralBody              Earth
EphemerisTimePos

      1.0000      11257327.374      -6787786.8  -3541540.4
      2.0000      9351815.651       -3729687.1  -1952861.1
      3.0000      8166117.553       -1829644.8  -966447.6
      4.0000      7544186.221       -832091.5   -450045.5
      5.0000      7244231.541       -348315.4   -201458.3
      6.0000      7106618.283       -122870.5   -87636.9
      7.0000      7045533.070       -18890.58   -37228.5
      8.0000      7019031.354       30331.166   -15428.3
      9.0000      7007709.781       55569.951   -6166.8
      10.0000     7002909.628       70523.637   -2286.4
      11.0000     7000863.699       81136.736   -678.4
      12.0000     6999963.580       89940.947   -18.2
      13.0000     6999532.475       97998.937   250.9
      14.0000     6999288.877       105751.622  359.9
      15.0000     6999116.800       113380.228  403.8
      16.0000     6998968.824       120958.657  421.4
      17.0000     6998825.694       128516.845  428.4
      18.0000     6998679.632       136066.847  431.2
      19.0000     6998527.513       143613.489  432.3
      20.0000     6998368.087       151158.694  432.7
      21.0000     6998200.856       158703.221  432.9
      22.0000     6998025.622       166247.365  433.0
      23.0000     6997842.307       173791.238  433.0
      24.0000     6997650.880       181334.879  433.0
      25.0000     6997451.329       188878.296  433.0
      26.0000     6997242.831       196450.645  452.5
      27.0000     6997026.065       204025.365  473.8
      28.0000     6996801.463       211587.806  487.0
      29.0000     6996568.981       219140.605  494.0
      30.0000     6996328.520       226687.795  497.3
      31.0000     6996080.008       234232.028  498.8
      32.0000     6995823.404       241774.722  499.5
      33.0000     6995558.689       249316.562  499.8
      34.0000     6995285.853       256857.862  499.9
      35.0000     6995004.891       264398.756  500.0

END Ephemeris
```